The AHI: An Audio and Haptic Interface For Simulating Contact Interactions

by

Derek DiFilippo

B.Sc Mathematics and Computer Science, McMaster University, 1998

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

Master of Science

in

THE FACULTY OF GRADUATE STUDIES

(Department of Computer Science)

We accept this thesis as conforming
to the required standard

The University of British Columbia

August 2000

In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Department of COMPUTER SCIENCE

The University of British Columbia
Vancouver, Canada

Date 28 AUGUST 2000

Abstract

A contact interaction occurs when two rigid objects strike, scrape, or slide against one another. Auditory and haptic (touch) feedback from contact interactions can provide useful information to an individual about their world. We have implemented a prototype human-computer interface that renders synchronized auditory and haptic contact interactions with very low (1ms) latency.

This audio and haptic interface (AHI) includes a Pantograph haptic device that reads position input from a user and renders force output based on this input. We synthesize audio in real-time by convolving the force profile generated by user interaction with the measured audio impulse response of the real-world version of the virtual surface. The resulting auditory and haptic stimuli are tightly coupled because we produce both using the same force profile. Also, because we use a dedicated DSP for haptic control and audio synthesis we are able to achieve negligible system latency. The AHI is the only human-computer interface that we know of for providing closely coupled auditory and haptic stimuli with guaranteed low latency.

Our work with the AHI draws on research results from a variety of fields: in haptics, audio synthesis, robotic measurement, and psychophysics. We have conducted a pilot user study with the AHI to verify that the system latency lies below the perceptual threshold for detecting synchronization between auditory and haptic contact events. We have also presented our work as a live demonstration at an international conference and have taken some preliminary steps toward integrating the AHI with a rigid body dynamic simulation. These three separate evaluations suggest that the AHI device and algorithms could prove valuable for further perceptual studies and for synthesizing continuous contact interactions in more general virtual environments with commercial haptic devices.

Contents

List of Tables

List of Figures

Acknowledgements

The following people made this thesis possible. They also made the entire process an enjoyable, educational, and revealing experience for me. I am grateful.

Dinesh Pai for being a constant source of enthusiasm and intelligence. Jacob Ofir and Josh Richmond for great advice and great company. Paul Kry for equally great advice, company, and his dynamic simulation code. Jochen Lang and Doug James for being ideal lab companions. Rod Barman for technical support. Valerie McRae for administrative support. Kees van den Doel for encouraging me to work with computer audio. Karon MacLean for excellent suggestions that significantly improved this document.

Curt Golden for being a catalyst. The Seattle Guitar Circle for musical support. The Seattle Repertoire Circle (Scott Adams, Chris Gibson, Stephen Golovnin, John Henning, Travis Metcalf, JT Milhoan, John Sinks, and Greg Sundberg) for giving me the opportunity to play my guitar again. Ed Reifel for inviting me to play Fracture at his recital.

Whitney Black for being my best friend. My parents, Nino and Judy, for their endless patience and goodwill.

DEREK DIFILIPPO

The University of British Columbia
August 2000

Chapter 1

Introduction

1.1 Overview

A person manipulating a physical object creates contact interactions. Our everyday lives are full of them: sliding a coffee cup across a table, tapping our fingers on a computer keyboard, etc. When we create these types of interactions we hear sounds and also feel the resulting forces in our hand. General contact interactions include continuous motions such as scraping and sliding as well as discrete contact events like tapping and hitting.

These contact interactions can produce characteristic sounds and forces that communicate valuable information about our relationship with physical objects and the surrounding environment. By using our ears and hands we can tell if the coffee cup was placed safely on the table or if the table is made of glass or wood. Depriving someone of this sort of feedback from their interactions could prevent them from effectively navigating and controlling their environment; or worse, could prevent them from experiencing the world to their fullest capacity.

An effective human-computer interface for interacting with a virtual envi-

ronment would allow the user to tap and scrape virtual objects in the same way we can tap and scrape real objects. Moving the interface handle as one would move a computer mouse, the user could interact with a force-feedback device coupled to a virtual probe as shown in Figure 1.1. As the probe bumps along the surface of the object, in addition to providing visual feedback, the interface would also create realistic auditory and haptic (touch) cues. These cues should be synchronized so that they appear perceptually simultaneous. They should also be perceptually similar – a rough surface should both sound and feel rough. This type of interface would improve the amount of control a user could exert on their virtual environment and also increase the overall aesthetic experience of using the interface.

A *multimodal* interface renders several synchronized perceptual modes based on user input to a virtual environment. In the example described above, the three modes are visual, auditory, and haptic. At present, to our knowledge, even the primitive multimodal interface described in Figure 1.1 does not exist: the auditory mode is the missing component. Force-feedback devices that allow for interaction with limited three dimensional graphical environments are commercially available, but without the capacity to represent and render appropriate sounds. This thesis presents a prototype multimodal interface for rendering synchronized sounds and forces based on user input.

1.2 Challenges

Our challenge is to develop and implement general algorithms and techniques for multimodal interactive simulations that balance the tradeoff between high realism and low latency. These competing design requirements make demands on the synthesis and control algorithms we implement and on the hardware devices that execute

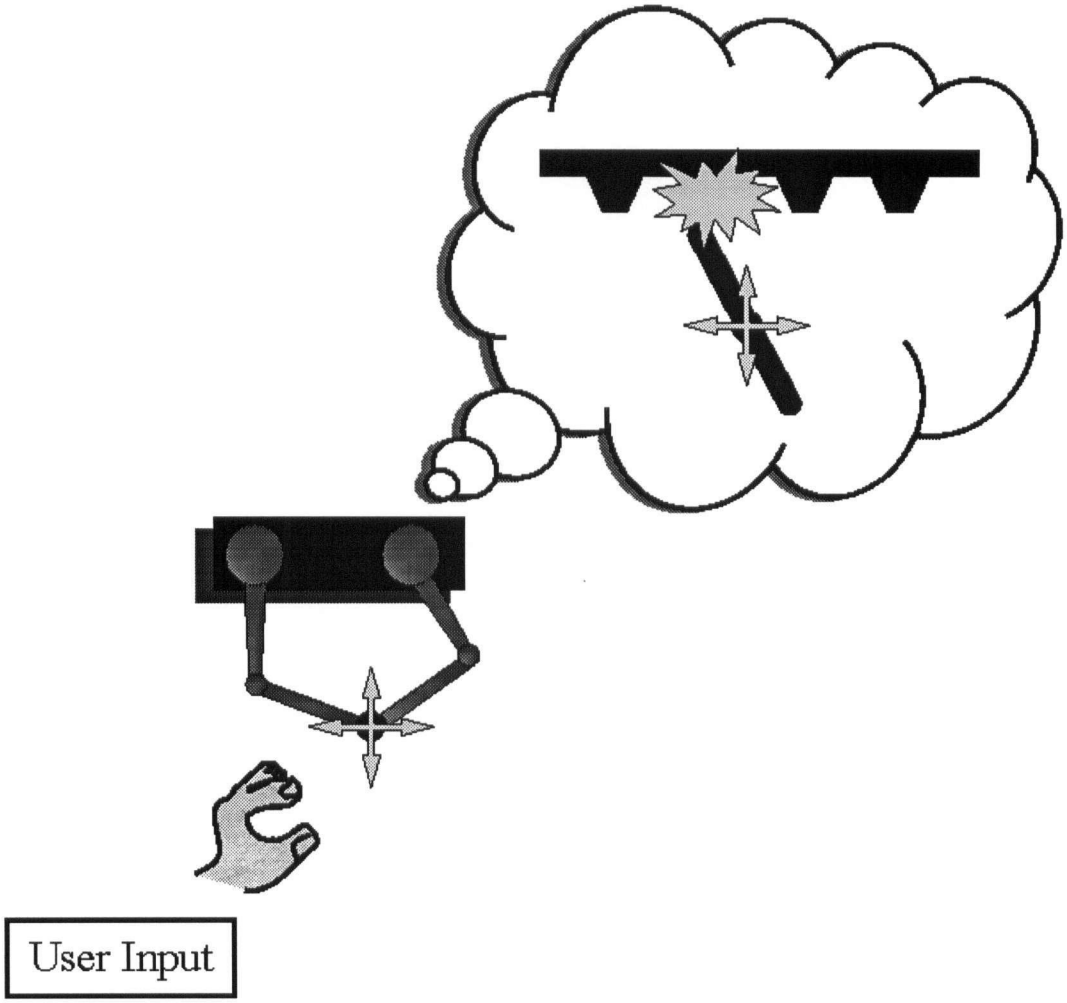

Figure 1.1: An example of a multimodal interface to a virtual environment. As the user manipulates the handle of the device, contact interactions occur between a virtual probe and surface. A multimodal interface should provide synchronized visual, auditory, and haptic feedback to the user.

the software. We focus here only on synthesizing contact interactions – in this thesis we do not consider the collision detection preceding the contact or the rigid body dynamics that may result from the contact.

Ideally, the synthesis algorithms we use will have some physically-based parameters. The auditory and haptic properties of a virtual object should be based on measurements of real objects whenever possible. For example, we might wish to use these virtual objects to represent remote objects to discerning user groups varying from NASA scientists to e-commerce customers. A robotic measurement system that can extract certain kinds of these "reality-based" models is being developed at UBC. We need algorithms that can render this new source of data and a multimodal interface that can help test the perceptual validity of the models.

At the same time, these physically-based algorithms must produce results quickly, perhaps in an anytime fashion, to satisfy the human perception of simultaneity between the actions they take and the feedback they receive from the device. The framework for simulating contact should be coupled with the underlying process of contact and collision to allow for sophisticated responses based on user interaction. If the user decides to suddenly scrape the virtual object, or to tap out a rhythmic pattern, the synthesized auditory and haptic stimuli should change appropriately and simultaneously – just as they would in a real environment. In addition, there are also hard real-time constraints to ensure that the haptic feedback remains stable – it is even possible that if these constraints are not met the user could suffer some injury.

A variety of haptic devices and a few audio synthesis techniques exist that could potentially balance the tradeoff between high realism and low latency for simulating contact interactions. However, there has been no widespread attention

given to the rendering of multimodal contact interactions like the one sketched in Figure 1.1, partially due to the limitations of widely available operating systems (e.g. Windows NT) in providing real-time interactive performance (for audio in particular). Our challenge is to implement an effective "proof of concept" device that is convincing enough to generate interest in multimodal simulation of contact interactions and to encourage system designers to consider using the algorithms that run our device in their own implementations.

1.3 Thesis Contribution

In this thesis we present our implementation of an experimental audio and haptic interface (AHI) for displaying sounds and forces with low latency and sufficient realism for interactive applications. The novelty of the AHI lies in the tight synchronization of the rendered auditory and haptic signals. User interaction with a simulated environment generates contact forces based on collisions with a flat or textured plane. These forces are rendered to the hand by a haptic force-feedback device and to the ear as contact sounds. This is more than synchronizing two separate events. Rather than triggering a pre-recorded audio sample or tone the audio and the haptics change together when the user applies different forces to the object.

Figure 1.2 shows the AHI feedback loop. The user manipulates the handle of a force feedback device by grasping it and moving it in the plane as one would move a computer mouse. The position of the handle determines a contact force that constrains the probe to the surface of a plane. This contact force is the basis of both a haptic signal (motors exert force on the handle) and an audio signal (coming out of the speakers). Using the same contact force to create auditory and haptic stimuli mimics how these stimuli are created in real contact interactions.

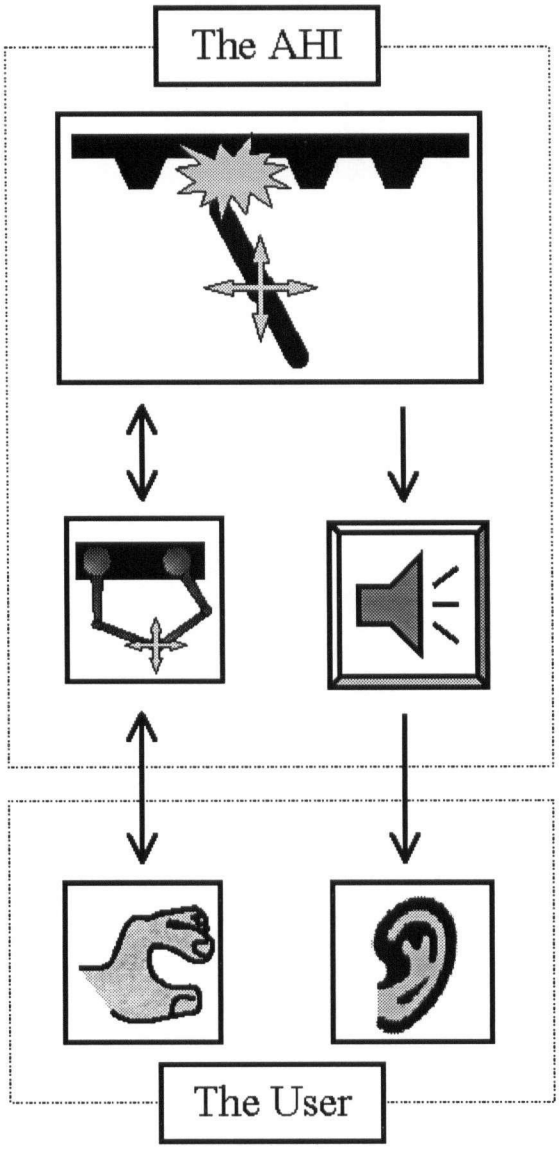

Figure 1.2: The AHI feedback loop. The user manipulates the handle of a force feedback device by moving it in the plane. This provides position input to the AHI simulation. If a contact occurs, the AHI returns force feedback to the user's hand through the haptic device, and auditory feedback to the user's ears through a set of speakers. Using the same contact force to create auditory and haptic stimuli mimics how they are created by real contact interactions. Current haptic devices do not provide the synchronized audio "channel" depicted in this figure.

We use an impulse response model for audio synthesis. Audio signals are computed as the discrete convolution of a measured audio impulse response with the contact force. In other words, this audio synthesis algorithm responds linearly to contact force. If we generate a rough contact force (say corresponding to a rough surface), the temporal information in the force profile is preserved by the audio convolution, and the resulting synthesized audio signal is also "rough." Section 2.3 outlines the suitability of our linear audio synthesis algorithm for this application.

We use a dedicated analog input/output (I/O) card with an on board digital signal processor (DSP) to capture user input and to display haptic forces and audio signals. As a result, we can bypass any scheduling or driver latencies that may be imposed by typical operating systems. Our total system latency for rendering synchronized auditory and haptic stimuli can be as low as 1ms. There is no published work that we know of describing a device that has similar capabilities.

Of course, building a device to a certain technical specification provides no guarantee of how a user will perceive its efficacy. However, the AHI could be useful to establish lower bounds for human perception of synchronized audio and haptics that give system builders some concrete design criteria to target. An analogy would be to the design of computer monitor hardware to support refresh rates of 60Hz, or higher. This refresh rate is well known to be sufficient for comfortable viewing, and sufficient to simulate continuous motion. Along these lines, we conducted an informal user study to confirm that the overall system latency for rendering integrated audio and haptic stimuli lies below the perceptual threshold for simultaneity between the two perceptual modes. We also presented a live demonstration of the AHI at a recent robotics conference to help informally evaluate the usefulness of the device, and to gauge public interest in multimodal contact interaction. Our most recent work has

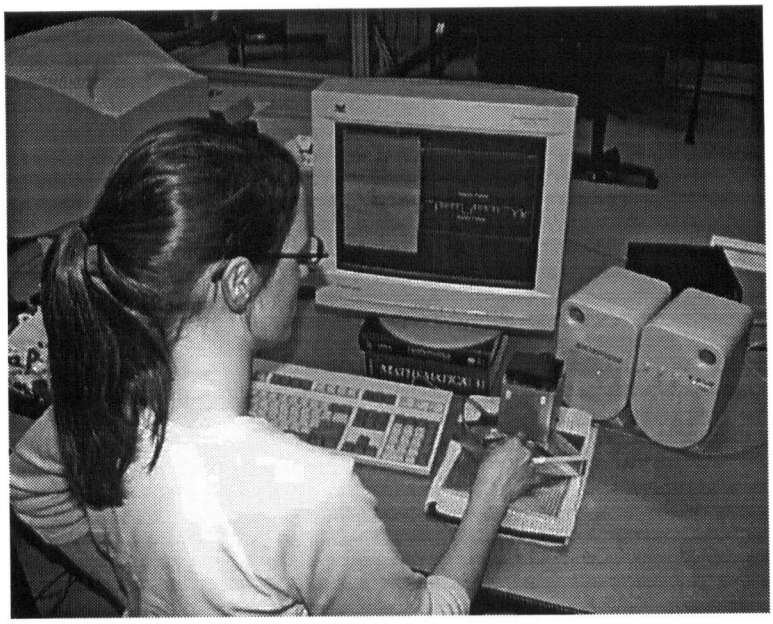

Figure 1.3: The AHI in its natural habitat, the Lab for Computational Intelligence. The user holds the AHI handle with her right hand. Audio speakers are positioned just behind the AHI device. The computer monitor shows a Java GUI for modifying simulation parameters and for viewing output signals.

been towards integrating the AHI with a rigid body dynamic simulation. The AHI allows the user to drag a convex object across a plane. The dynamic simulation updates the graphics according to physical constraints and user input. The AHI renders accompanying sounds and forces.

Figure 1.3 shows the AHI in its natural habitat. The user grips the handle with their right hand and moves it in the plane as they would move a computer mouse. The left side of the monitor screen shows a Java graphical user interface (GUI) for loading sound models and controlling interaction parameters and the right side shows a graphical window for viewing haptic forces and audio signals in real-time. Audio speakers are positioned just behind the AHI device.

Our work on the AHI draws on research results from a variety of fields: in haptics, audio synthesis, robotic measurement, and psychophysics. The AHI is an example of how current research directions combined with specialized hardware can help point the way to the next generation of human-computer interfaces for multimodal interaction.

1.4 Guide to the Thesis

Chapter Two reviews related work done in haptics, audio synthesis, robotic measurement, and also covers some perceptual issues and their implications for designing interactive systems. Chapter Three describes the AHI hardware for user input, audio synthesis algorithms, simple contact models for calculating haptic and audio contact forces and the design of the control code for rendering auditory and haptic stimuli. Chapter Four presents three evaluations of the AHI based on an informal user study, results and comments from a three-day live conference demonstration, and preliminary steps towards integrating the AHI with a rigid body dynamic simulation. Chapter Five summarizes the work done in this thesis and suggests some future research directions.

Chapter 2

Related Resarch

2.1 Overview

This chapter will review related research in haptics, audio synthesis, robotic measurement, rigid body dynamic simulation, and psychophysics. We will overview each research area to motivate and clarify the design choices behind the AHI.

Our goal in developing the AHI was to implement a device that could be precise enough for use in perceptual studies and interesting enough to motivate future research in simulating contact interactions. The sections in this chapter on operating systems and audio latencies describe the current state of the art in rendering real-time audio on widely available operating systems. We will see that it is only recently that these operating systems have begun to approach sufficiently low latencies; however, there are only a few empirical studies of how low of a latency is required to satisfy the perception of simultaneity between auditory and haptic stimuli.

As a result of these operating system audio latencies, and their uncertain perceptual consequences for synchronized audio and haptics, the central design cri-

terion for the AHI has been to maintain a low and consistent system latency without compromising the quality of the simulated contact interactions. To satisfy this design criterion, we have selected a particular set of hardware and software. We control our custom-built haptic device with a dedicated DSP and I/O board. We use an audio synthesis algorithm that responds in real-time to contact forces and can be parameterized using automated robotic measurements. The first two sections of this chapter will detail related research in haptics and audio synthesis with the goal of clarifying our particular design choices.

2.2 Haptics

The word *haptics* is derived from the Greek, and means related to or based on the sense of touch. Human haptics encompasses all components of the control system that connects the hand to the brain. This includes the mechanical properties of the hand, the sense of static touch at the fingertips, kinesthetic information based on the position of the limbs, and the cognitive processes that respond to sensory input [Srinivasan and Basdogan, 1997]. Through haptic senses, people are able to manipulate their world and receive information about the objects in that world.

In the last 10 years the field of computer haptics has grown immensely. Parallel to the field of computer graphics which represents and renders visual phenomena, computer haptics looks to represent and render haptic stimuli to a user. Computer graphics currently boasts a variety of specialized hardware and sophisticated algorithms that are widely available on home personal computers (PCs). We are all familiar with the computer monitor, an ubiquitous graphic interface. It it hard to imagine using a computer without one. Similarly, the computer mouse is an indispensable tool for interacting with a computer. The mouse allows us to use our

hands to manipulate objects in a natural way by pointing, clicking, and dragging. This kind of interaction uses only one half of the potential for haptic interaction – it changes objects based on position input. What the mouse does not provide is force feedback based on this input. If you drag your file to the edge of the computer screen, the mouse does not provide any haptic feedback to tell you that this collision has occurred.

The most common type of *haptic interfaces* provide force feedback based on position input from a user. Several commercial devices exist that support haptic interaction. The most popular one in the research community is the PHANToM by Sensable [Massie and Salisbury, 1994]. It (like most haptic devices) is an impedance device, reading 6 degrees of freedom (DOF) in position input from the user and rendering 3 DOF force feedback to the user's hand (Figure 2.1). The PHANToM has a well-established user base and produces high fidelity haptic stimuli but is priced far beyond what the typical computer user can afford. Haptic mice such as the Logitech Wingman Mouse, and haptic joysticks such as the Microsoft Sidewinder, have been on the market for a few years [Logitech, 2000, Microsoft, 2000]. These consumer devices have fidelity roughly proportional to their price. This recent expansion in the availability of haptic devices means that novel improvements could be quickly assimilated by the haptic community and passed on to current users.

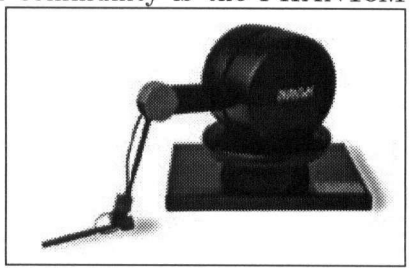

Figure 2.1: The PHANToM force feedback device [Sensable, 2000].

The wide availability of haptic devices also motivates and enables some novel psychophysical experiments based on haptic interaction. Katz was a psychologist who wrote an influential book about the world of touch [Katz, 1989]. He worked

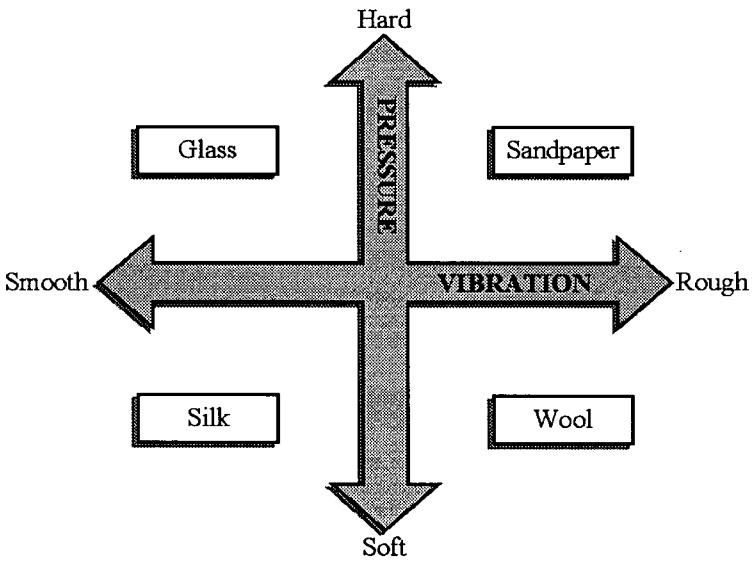

Figure 2.2: Pressure and vibration axes, after [Katz, 1989]. Common materials (glass, sandpaper, silk, wool) are plotted along the two axes. The pressure axis characterizes hardness properties and the vibration axis characterizes roughness properties.

with the hand as the primary organ of touch. His work emphasized the importance of movement as a formative factor in tactual haptic phenomena: "as well-nigh as indispensable for touch as light is for color sensations" (pg. 76). For example, if you place your hand on a flat surface you may not perceive the texture of the surface until you stroke your hand across it. This movement creates a vibration sense and Katz believed that the vibration sense is independent of and superior to the pressure sense. Figure 2.2 shows pressure and vibration axes with some common materials plotted against these two dimensions.

General contact interactions generate dynamic vibration stimuli – both audio and haptics. Haptic interfaces both demand and facilitate research into the perception of dynamic vibration. Recent work on perceiving roughness using a rigid

probe reveals that dynamic stimulus properties are more relevant than when using the bare finger [Lederman and Klatzky, 1999]. Current haptic devices are capable of providing dynamic vibration stimuli based on user interaction. It is possible to represent a textured surface as a collection of polygonal patches, or by applying procedural textures inspired by Gaussian surface deviations [Siira and Pai, 1996, Fritz and Barner, 1996]. The precise control afforded by computer controlled haptic interfaces allows for systematic and repeatable variation of stimulus properties, and also for automatic measurement and logging of experiment variables such as contact force and velocity.

The haptic device for the AHI is a Pantograph, based on a design by Hayward, and custom built for our laboratory [Ramstein and Hayward, 1994]. The Pantograph is well known in the haptics research community. Our particular Pantograph does not require operating system support like the PHANToM. By using our own control hardware we can reliably synchronize the haptic output of the Pantograph and our synthesized audio. Currently, this is not possible with the PHANToM and Windows NT, as will be discussed in Section 2.7. A Canadian company, Haptech, has commercialized a less expensive and lower performance version of the Pantograph technology and is aggressively marketing its device as a high fidelity haptic mouse [Haptech, 2000]. Using the Pantograph for the AHI gives us the fidelity to reliably render high quality stimuli and allows us to synchronize our haptics and audio.

2.3 Audio Synthesis

From the earliest days people have been using computers to produce musical sounds. Some of the first efforts were inspired by simple analog circuits that mimicked sound

generation in musical instruments [Roads, 1996]. By the early 1980's a variety of hardware synthesizers were commercially available and in wide use. The Musical Instrument Digital Interface (MIDI) protocol for communicating control data between computers and synthesizers allowed for the creation of software sequencers for writing and performing music directly on a PC [Anderton et al., 1994]. In the past decade, as CPU speed and soundcard capabilities have increased, the home PC has blossomed into a stand alone recording studio and music machine for many.

One aspect of audio synthesis and control that has not received the same amount of attention as musical applications is that of synthesizing everyday environmental sounds based on contact interactions. William Gaver was one of the first to discuss the importance of *everyday listening*: "the act of gaining information about events in the world by listening to the sounds they make" [Gaver, 1988]. It is everyday listening that tells us that our coffee cup has been placed on the table or that we are in a church rather than a cubicle. In addition to performing some user studies investigating the perception of environmental sounds, Gaver developed a specification for creating auditory icons, similar to the visual icons we have on our computer desktop. He wanted these auditory icons to be parameterized based on object properties such as age, size, or speed [Gaver, 1993].

Gaver's requirements for auditory icons are related to the ones we have for the AHI. Our synthesized sounds need to

- be generated in real-time based on user interaction,
- respond to continuous input data, such as contact force,
- represent the auditory properties of everyday objects,
- be parameterizable based on measurements of everyday objects.

For a simulation driven by discrete events, recorded samples can satisfy some of these requirements. If we want to incorporate the sound of a struck bell into our simulation, we can strike a bell, record it, and store it in memory on our computer. This representation is model-free or unstructured – sound data is represented and rendered as a raw signal. For example, a Windows operating system can trigger different recorded samples (and synthesized sounds) when a user clicks on an icon or saves a file. But what if we wish to change the size or material of the bell? Or if we want to change where the bell is struck, or worse, scrape the bell with our fingers? Storing a recorded sample for each one of these events becomes prohibitive. Model-free representation at this temporal scale cannot respond to continuous interactions based on user input.

One can imagine, however, that if the recorded samples were short enough they could be concatenated in response to continuous interactions. User interaction would shape a stream of pre-recorded sound *grains* of short duration. By recording the sound of a scraped bell and splitting it into grains, it is possible to synthesize continuous sounds by concatenating these grains indefinitely. Stretching or shrinking the grain length also scales the frequency spectrum of the resulting signal. This model-free representation is commonly known as *granular synthesis.* There is no explicit model of the object. Like wavelet representations, a granular synthesis technique represents and renders signals using a basis that decomposes signals into time-frequency grains. Shaping and modifying these grains allows for arbitrary time-frequency modifications of an original signal.

Wavelet and granular synthesis techniques have been investigated for synthesizing continuous textured sound based on user interaction [Roads, 1996]. The first real-time implementation of granular synthesis required dedicated hardware

and complicated parameter control techniques [Truax, 1988]. Controlling a granular synthesizer requires specifying local parameters such as grain length, grain amplitude, and grain pitch, as well as global parameters such as grain rate and grain overlap. The tremendous amount of flexibility afforded by granular synthesis appeals to electroacoustic composers. Truax's work focused on this application and did not address how to use the technology for everyday sounds.

More recent studies have considered using wavelet techniques for synthesizing environmental sounds and continuous textures [Miner et al., 1996]. They report good results for synthesizing diverse sounds such as rain, car motors, breaking glass, and jazz themes [Bar-Joseph et al., 1999]. Wavelet signal decomposition approaches are appealing because they are amenable to powerful analysis/resynthesis techniques as well as real-time implementations based on the inverse wavelet transform. Some unanswered issues for wavelet techniques are how to map physically-based variables to resynthesis parameters to create convincing audio signals and how to efficiently build and manage a library of basis grains.

These unanswered issues, as well as our central design criterion of low system latency, led us to consider the limit where there is no temporal information in the audio representation. If our haptic contact interaction parameters are changing at a rapid rate (typically 1kHz), our audio representation must be able to respond to these changes to maintain tight synchronization. Recorded samples with arbitrary duration or even wavelet grains on the order of tens of milliseconds will not satisfy our design requirements. We must decompose the synthesis model into two parts: a static frequency content (pressure) that is specified prior to any interaction and a dynamic temporal content (vibration) generated by real-time interaction. Figure 2.3 shows the parallel representation to Figure 2.2 for auditory roughness and hardness.

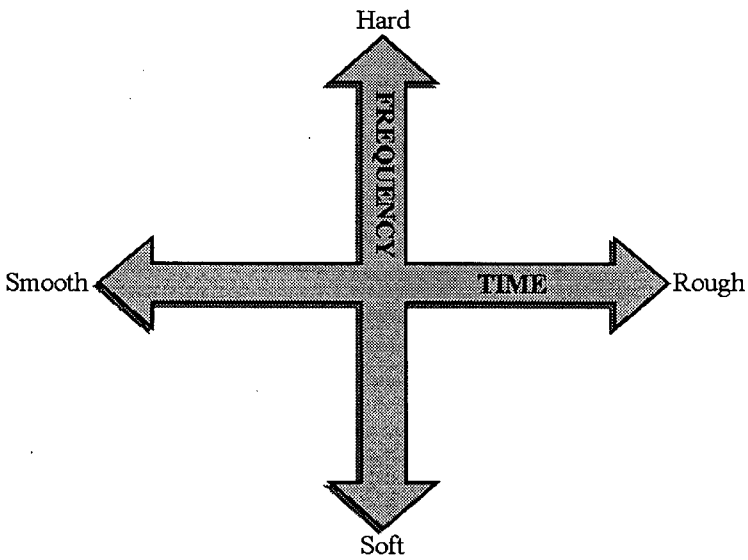

Figure 2.3: Time and frequency axes. Analogous to the vibration and pressure axes of Figure 2.2, here the time axis characterizes hardness properties and the frequency axis characterizes roughness properties. We decompose the audio synthesis model into two parts: a static frequency content (pressure) that is specified prior to any interaction and a dynamic temporal content (vibration) generated by real-time interaction.

Following Gaver's initial proposal, Kees van den Doel developed an *audio impulse response* synthesis model that satisfies the four main requirements listed at the beginning of this section [van den Doel and Pai, 1998]. A weighted sum of damped sinusoids represents the audio impulse response at a particular point on an object. The only temporal information that this model contains is the damping rate of each sinusoid. All other dynamic temporal information comes from the input to the model: for our purposes, we consider this as a force input. The synthesized audio signal is the discrete convolution of the audio impulse model with input forces. The linearity of the model satisfies two of our four criteria: it responds dynamically to input force and it responds in real-time to input force. Audio impulse

models can represent sophisticated contact interactions with rigid everyday objects [Cook, 1997]. Finally, the model can be parameterized based on measurements of everyday objects.

2.4 Robotic Measurement

Reality-based modeling is one of the newest areas that has been gaining momentum in computer modeling. In ecommerce, computer games, film production, telerobotics, and telemedicine, it is becoming increasingly more important to represent and render virtual objects that are based on real objects. To be effective, these reality-based models should move beyond a visual representation of an object to include representations of how the object sounds and haptic properties such as texture and deformation.

The UBC ACtive MEasurement facility is an experimental robotic platform for obtaining sophisticated reality-based models of small everyday objects [Pai et al., 1999]. There is a motion control platform for positioning the target object, a stereo camera for obtaining depth images, and a robotic arm that can apply forces. The whole facility can be controlled over a network by writing Java programs that execute a particular series of measurements on the object. This computer control allows for precise and repeatable measurements to be taken over the surface of the object.

Figure 2.4 shows a high level overview of how the AHI project integrates with ACME. In the broadest terms, ACME provides the parameters for our reality-based models and the AHI is one device that uses these models to synthesize contact interactions. ACME does not currently obtain full object models and the AHI does not render full object models; however, audio impulse response models as described

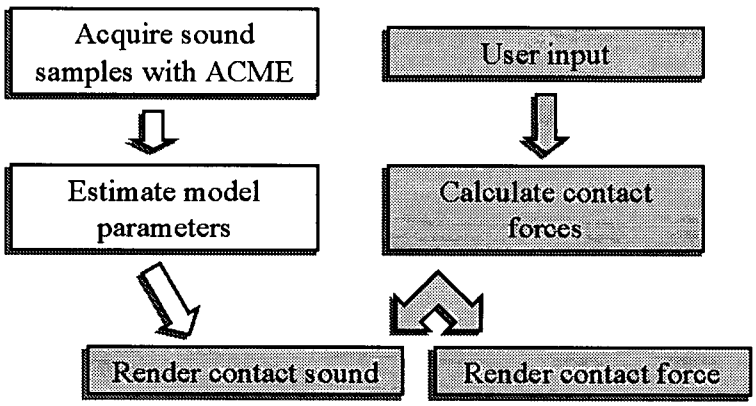

Figure 2.4: An overview of how the AHI project integrates with the ACME facility. ACME provides parameters for our reality-based models (off-line) and the AHI is one device that uses these models to synthesize contact interactions (on-line).

in Section 2.3 can be acquired [Richmond 2000]. ACME can provide us with a list of audio synthesis parameters – amplitude, frequency, and damping – associated with a multiresolution surface representation. We do not currently use the position information, but we do use these acquired sound models in the AHI simulation.

2.5 Rigid Body Dynamic Simulation and Sound Synthesis

Rigid body dynamics can be an essential part of a multimodal interaction. If we strike a virtual ball, we should see and hear it roll away from us. Or if we grasp this ball and slide it across a surface, we should receive appropriate feedback. Kry et al. have developed a rigid body simulation that evolves the forward dynamics of a single contact point between two smooth surfaces [Kry and Pai, 2000, Pai et al., 2000]. Their simulation exhibits rolling and sliding behaviours based on geometry, gravity,

and friction. In addition, a user can initiate these rolling and sliding behaviours based on either mouse input or by using the AHI handle to provide position input.

We would like to use the resulting physically-based simulation parameters (linear and angular velocities, for example) to generate appropriate force profiles for input to our audio impulse response model. This would allow us to incorporate a visual dynamic simulation with the AHI's haptics and audio. The missing component is the real-time (or near real-time) synthesis of believable scraping and rolling sounds.

The majority of the research on rolling sounds has been conducted by engineers on wheel/rail noise interactions for trains [Thompson and Jones, 2000]. In general, the literature compares measured frequency spectra of real trains with frequency domain models of noise produced by surface roughness and wheel/rail geometry [Remington, 1987]. Useful models suggest new materials and geometries to minimize the total sound pressure level – no consideration is given to real-time synthesis of particular rolling sounds in the time domain.

Hermes describes an input force model for simulation of scraping and rolling sounds using an audio impulse response model [Hermes, 1998]. He suggests using *Poisson pulses* – impulses with exponentially distributed interarrival times – as force input for sound synthesis. Varying the mean of the distribution gives some control over the roughness of the output signal. Satisfactory scraping sounds can be synthesized in this manner.

He considers two additional filtering stages to produce rolling sounds. First, rolling impulses should have a more gradual onset rather than the sharp exponential decay of a standard impulse response. Poisson pulses filtered by an impulse response sound too "ticky" to be perceived as rolling. *Gammatone* filters are time-invariant

impulse response models that have a gradual onset [Slaney, 1993]. Second, rolling impulses should be amplitude modulated with both the frequency and maximum amplitude as a function of angular velocity. Amplitude modulation of this sort is reported to increase the perception of rolling. The perceptual validity of these synthesis variables has not been firmly established. Houben and Hermes have conducted experiments using recorded samples of real rolling balls which suggest that subjects are able to identify the size and velocity of rolling balls based only on the auditory stimulus [Houben et al., 1999]. Repeating this experiment with synthesized signals would help identify perceptually relevant force generation and filtering stages.

2.6 Cross-Modal Similarity and Synchronization

How realistic must a computer simulation be in order to be useful or believable, or both? The computer graphics community has invested 30 years of hardware and software development to bring the state of the art to the enviable position it is in now – and development continues at a rapid pace. The most fundamental constraint for convincing continuous motion is the refresh rate. Standard film projectors have 24 still pictures per second and interlaced raster scans for CRT's start at a minimum of 30Hz [Hochberg, 1986]. Similar constraints are required for audio and haptics. The upper limit of human hearing and Nyquist's sampling theorem demands that CD's have a sampling rate of 44.1kHz [Steiglitz, 1996]. The haptics community has converged on a refresh rate of 1kHz for stable simulations [Massie and Salisbury, 1994]. These graphic, audio, and haptic refresh rates are well-established standards based on human perceptual limits.

Much less is known about the perceptual requirements for synchronization and similarity between the three sensory modalities (graphic, auditory, haptic) that

are most important for computer simulations. The McGurk effect is an early example of how a lack of similarity between recorded audio and video could affect the observer's perception [McGurk and MacDonald, 1976]. In McGurk's experiment, observers were presented with spoken words and simultaneous video of a human face speaking similar, but different, monosyllables. The observers tended to hear a sound somewhere between the visual and auditory stimuli. This result is an example of how a mismatch in similarity can cause ambiguity.

Some researchers in coupled haptics and audio have intentionally created a mismatch in similarity. Their hope is that a "hard" audio signal will increase the user's perception of haptic stiffness and therefore decrease the necessary active stiffness rendered by the haptic device. In their experiments, pre-recorded audio samples or tones were triggered by contact events and were not synthesized based on contact force. In two separate studies in different labs, one claimed that "the auditory stimulus did not significantly influence the haptic perception" [Miner et al., 1996], and the other found that "sound cues that are typically associated with tapping harder surfaces were generally perceived as stiffer" [DiFranco et al., 1997]. These studies suggest that coupling audio and haptics could help create more sophisticated perceptions of solidity, shape, location, and proximity.

How synchronized do haptic and audio events have to be in order to be perceived as a single simultaneous event? Imagine playing a MIDI keyboard that triggers sounds on your PC. If the sound arrives 10 seconds after you depress a key, the two events would definitely be perceived as separate (or even unrelated). A 3 byte MIDI note message takes $0.96\mu s$ to transmit, and if it triggers a hardware synthesizer with essentially zero latency, the haptics and audio will be perceived as a single event [Anderton et al., 1994]. The synchronization tolerance for coupled

audio and haptics lies somewhere above $1\mu s$ and somewhere (well) below 10 seconds.

Playing any musical instrument is an exercise in coupling audio and haptics. Rasch developed a technique for measuring and describing asynchronies in performed ensemble music [Rasch, 1979]. Professional woodwind, string, and recorder players performed polyphonic scores. "The data showed that asynchronization defined as the standard deviation of differences in onset time of simultaneous notes has typical values of 30 to 50ms." Although there is some variability in defining onset time, particularly for bowed instruments, these figures were found to be valid over a variety of tempi and instruments. Rasch reported that these professional musicians gave the impression of perfect synchronization, which suggests that his reported range of asynchronies are close to the minimum one could expect a human user to detect.

Levitin et al. have done work on the perception of auditory and haptic simultaneity that coheres with Rasch's results [Levitin et al., 2000]. In their study, subjects manipulated a baton. They would strike a horizontal surface (containing a capacitor) with this baton. By tracking position, velocity, and acceleration, the experimenters could predict the time of actual impact. Using these predictions, a digitized sample of a stick striking a drum was played back to the subject at random temporal offsets varying between -200ms and 200ms.

Subjects were asked to judge whether the baton strike and the audio sample occurred at the same or different times. The threshold where subjects considered the stimuli to be synchronous 75% of the time corresponded to -19 and 38ms; that is, the interval between the audio preceding the haptics by 19ms, and the audio lagging the haptics by 38ms. When adjusted for response bias (using confidence ratings) the corrected thresholds for detecting synchrony are -25ms and 66ms.

2.7 Operating Systems and Audio Latency

This section will review the current state of low latency audio on two widely available operating systems: Windows and Linux. In general, we define end-to-end system *latency* as the time for system input, CPU processing, and system output. For implementing the sort of real-time audio synthesis used in this thesis on Windows and Linux, the majority of overall system latency comes at the output stage and is due to the operating system. Operating system researchers face the challenge of generating near real-time response from a multithreaded paradigm designed for very general applications. The good news is that Linux systems currently report numbers in the 10ms range and the recent release of Windows 2000 and a new version of Direct Sound has greatly improved audio performance over NT4 and Win9x.

The Lab for Computational Intelligence has a PHANToM haptic device. It has been integrated with Java 3D graphics to create an interactive elastic modeling application [James and Pai, 1999]. Given the success of this application, and the allure of 3D graphics, it was natural to think of adding sound to a PHANToM application that included rigid bodies. We investigated this and found enough CPU power left to spare for audio synthesis. Also, the GHOST API for generating PHANToM haptic scenes simplifies the coding effort. The only obstacle to providing high quality low latency audio came from the Windows NT4 audio API and from sound card device driver interrupt scheduling conventions. Driver interrupts are triggered every 62ms for transferring computed audio buffers from the system memory to the sound card – this may approach acceptability [Creative, 2000] [personal correspondence]. However, the Direct Sound implementation for NT4 has a hardcoded audio buffer length of 8192 bytes. Audio will not come out of the speakers until this buffer is filled. At the highest possible sampling rate of 44.1kHz, this leads to

8192/44100 $\approx$ 186ms of latency. Since Sensable only supports the PHANToM on Windows NT4 and SGI IRIX (and because of separate requirements to maintain a cutting edge Java environment) we decided to postpone sonifying the PHANToM until newer drivers for either Linux or Windows 2000 were available.

Sound Lab (SLAB), a software-based system for interactive spatial sound synthesis, provides a recent benchmark for the performance of Windows 98 for an application that is structurally similar to the AHI [Wenzel et al., 2000]. SLAB is implemented using C++ and runs on an Intel Pentium III laptop. SLAB filters a steady tone in real-time based on virtual room size and head position of the user. A head tracker reads position and orientation of the head. Based on this input the steady tone is IIR and FIR filtered according to reverberation and head-related transfer functions. The output is returned to the user via headphones. The internal system latency for SLAB is 24ms and the majority (>90%) comes from the Windows 98 Direct Sound audio API [Wenzel et al., 2000] [personal correspondence]. To achieve even this amount of latency it is necessary to have a busy wait constantly polling the Direct Sound output buffer. Additional CPU cycles are lost during the busy wait.

Brandt and Dannenberg derived performance measurements of Windows NT4 and Windows 95 [Brandt and Dannenberg, 1998]. On NT4, they varied the CPU load and measured the actual time between 5ms callbacks. The callbacks and the competing *cpugrab* process were run at normal priority levels. As one would expect, average excess latency over 5ms increases with load. At 0% CPU usage the worst case latency was 15ms and at 70% it was 126ms. These are latency measures that any general NT4 application can expect. Unfortunately, as described previously, the hardcoded buffer length in NT4's audio system increases the au-

dio latency to 186ms [Brandt and Dannenberg, 1998]. On Windows 95 the same test using multimedia timers and native Direct Sound resulted in uniformly poor worst-case latency with a minimum of 123ms. Brandt and Dannenberg conclude that audio latencies on NT4 are limited by its audio system and on Windows 95 by non-deterministic user mode scheduling.

In addition to scheduling latencies, it is also necessary to consider interrupt latencies. Cota-Robles and Held compared Windows Driver Model (WDM) latencies for Windows NT and Windows 98 [Cota-Robles and Held, 1999]. They concluded that "for real-time applications a driver on Windows NT4 that uses high, real-time priority threads receives an order of magnitude better service than a similar WDM driver on Windows 98." Their results show that the best case interrupt latency on either OS is about 1ms and the worst case on Win98 is above 100ms.

To our knowledge, there are no thorough studies of audio latency on Windows 2000. The closest to a review of Windows 2000 that we have found comes from Twelvetone Systems [Kuper, 2000]. They sell a popular audio and MIDI sequencer called Cakewalk and therefore have a vested interest in low latency audio for increased performance of their audio mixing and MIDI recording and playback. In February of 2000, they stated on their webpage that "we believe an obtainable target for audio latency under Win2k is 5ms, even under heavy system loads."

This 5ms figure has also been reported for Linux systems. Various kernel tweaks are necessary to reach this performance – any kernel routine that takes several milliseconds before returning control to the scheduler will push this 5ms figure higher. Other factors that can push up latency levels include any motherboard that obtains exclusive access to the PCI/ISA bus during disk I/O and older PCI graphics cards which keep exclusive access to the bus for their own performance

	Latency (ms)	Limitation (and reference)
Windows NT4	186	buffer length [Brandt and Dannenberg, 1998]
Windows 95	123	user scheduling [Brandt and Dannenberg, 1998]
Windows 98	24	Direct Sound API [Wenzel et al., 2000]
Windows 2k	< 10	system load [Kuper, 2000]
Linux	< 10	system load [Senoner, 2000]

Table 2.1: A summary of expected audio latencies on Windows and Linux, as well as the dominant limitation for lowering these latencies. Recent operating systems have improved, but do not guarantee hard real-time performance.

reasons [Senoner, 2000].

2.8 Chapter Summary

This chapter has presented related research to motivate the design choices behind the AHI. Research in computer haptics and audio synthesis has provided the necessary components for the AHI's multimodal synthesis, but current operating systems cannot support the low latency we need to synchronize the auditory and haptic modes. By using specialized hardware we have the opportunity to design a multimodal interface that reliably performs within human tolerance for perceptual synchronization between contact events. The AHI's algorithms should be useful for synthesis on more general systems and for further psychophysical studies investigating more complex dynamic vibration phenomena. Chapter Three describes real-time audio and haptic simulation with the AHI.

Chapter 3

Real-time Audio and Haptic Simulation

3.1 Overview

This chapter describes real-time audio and haptic simulation with the AHI. A three degree of freedom Pantograph haptic device reads user input as position. Contact forces are generated based on user input. The AHI renders these forces as haptic feedback to the user's hand and as audio feedback to the user's ear. Our audio synthesis algorithm was developed at UBC by a previous student of Dr. Pai's [van den Doel and Pai, 1998]. Modifications to this algorithm for synthesizing rolling and scraping sounds are based on work by Hermes [Hermes, 1998]. For forces normal to the plane the haptic force synthesis equation we use is a standard penalty method based on a spring, damper, and impulse combination. We have also implemented some friction models to produce tangential forces. Real-time simulation of sounds and forces runs on a dedicated DSP using precisely timed interrupt routines.

3.2 Hardware

User input to the AHI comes from a 3 degree of freedom (DOF) Pantograph device, shown in Figure 3.1. The 5-bar mechanism is based on a design by Hayward [Ramstein and Hayward, 1994] but was extended to 3 DOF to our specification. It reads 3 DOF of position as user input and renders 3 DOF of forces as output. The user can move the handle in the plane (as one would move a computer mouse) as well as rotate the handle. There are two large Maxon motors attached to the base of the Pantograph which apply forces on the handle via the 5-bar linkage [Maxon Motors, 1996]. A small Maxon motor in the handle can exert a torque on the handle as well. The device, therefore, is complete for rigid motions in the plane, i.e., it can render the forces and torque due to any contact with a rigid body attached to the handle in a planar virtual world ("flatland"). We do not currently use the third rotational DOF for our work with the AHI.

There are rotational potentiometers attached to the large motor shafts. They are supplied with a known voltage. The base joint angles are controlled to be a linear function of potentiometer output voltage. A digital encoder measures the rotation of the handle. Both voltages and the encoder counts are input to a dedicated motion control board connected to a PC running Microsoft NT. The motion control board (MC8, Precision Microdynamics) has 14 bit analog to digital converters (ADCs) for reading the potentiometer voltages as well as quadrature decoders for reading the handle rotation [Precision MicroDynamics, 1999].

The processor on the MC8 board is an Analog Devices 21061 digital signal processor [Analog Devices, 1995]. It has a clock rate of 40 MHz and floating point hardware. All control code for synthesizing haptic forces and contact sounds executes on this DSP. Analog Devices provides a GNU licensed cross complier that

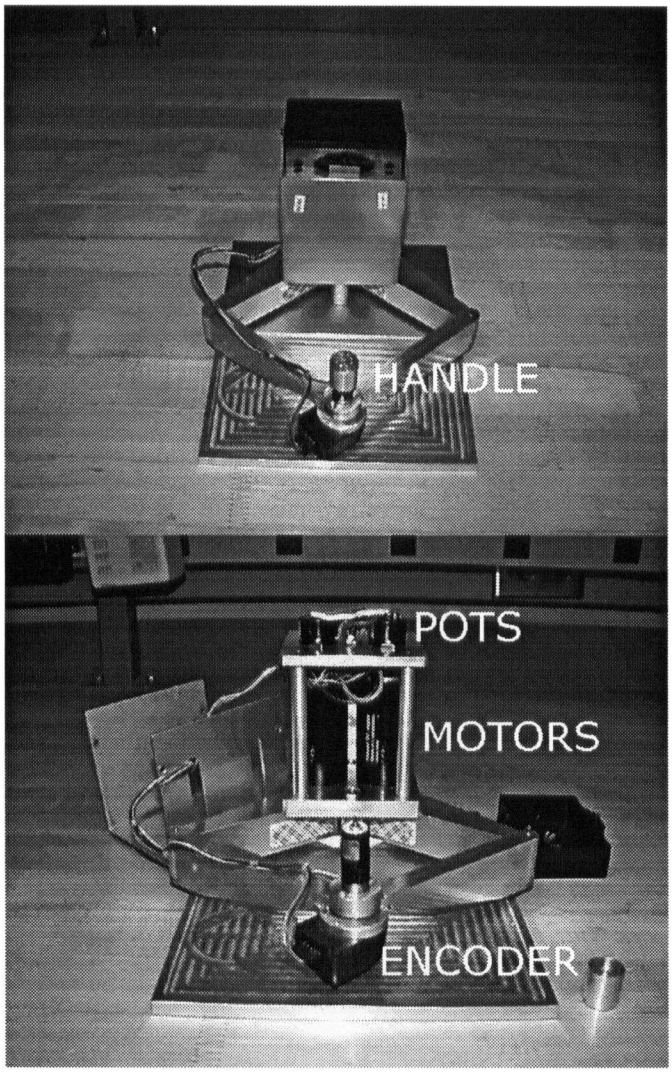

Figure 3.1: The Pantograph haptic interface. The handle, potentiometers, motors, and encoder are labeled.

compiles C code into executables for loading onto the DSP. Output voltages for controlling the Pantograph motors and for rendering audio are sent out through 14 bit digital to analog converters (DACs). Two Copley 306 PWM Brushed Servo Motor amplifiers provide currents to the large Maxon motors [Copley Controls, 1994]. The audio waveforms can be input directly into an amplifier and speakers; in our setup they are sent to the soundcard of the host computer for ease of capture and playback.

By using this specialized hardware, we bypass the complications that arise from balancing the needs of real-time deterministic response and ease of access from user-level software on a widely available operating system such as Windows NT. As discussed earlier in Section 2.7, kernel-mode Windows NT perhaps has the scheduling and interrupt latency to support the 1kHz control rate demanded by haptic applications, but definitely does not have the audio system to keep up with this control rate.

The AHI control code is compiled for the DSP and has exclusive control over its resources. This allows us to precisely time our control algorithms as well as accurately diagnose inefficiencies and bugs. In particular, we can achieve a consistent overall system latency of 1ms for synchronized changes in audio and haptics. We will define system latency and asynchrony in Section 3.7.

3.3 Audio Synthesis

In this section we describe the impulse response model we use for audio synthesis. We have extended this model by implementing Gammatone impulse responses driven by amplitude modulated Poisson pulses for synthesizing rolling and scraping sounds.

3.3.1 Impulse Response Model

We wish to simulate the audio response of everyday objects made out of wood, metal, ceramic, etc. Contact with these objects can be characterized by impulsive excitation of relatively few exponentially decaying, weakly coupled sinusoidal modes. Modal synthesis and impulse generation techniques have been developed for these types of percussive sounds [Cook, 1997]. We use the modal audio synthesis algorithm described by van den Doel, and discussed in Section 2.3. This algorithm is based on vibration dynamics and can simulate effects of shape, location of contact, material, and contact force. Model parameters are determined by solving a partial differential equation or by fitting the model to empirical data using the ACME facility [Richmond and Pai, 2000].

The sound model assumes that the surface deviation y obeys a wave equation. For simple geometries and materials the wave equation has an analytic solution, shown in equation 3.1. Typically μ is equal to a sum of eigenfunctions, Ψ. The ω^2 are the eigenvalues and are related to the vibration frequency of the object.

$$\mu(\mathbf{x},t) = \sum_{n=1}^{\infty}\Big(a_n \sin(\omega_n ct) + b_n \cos(\omega_n ct)\Big)\Psi_n(\mathbf{x}) \tag{3.1}$$

The resulting sum is undamped. This does not model real objects very well – once struck, the object would radiate sound forever. By adding a material dependent decay coefficient to the wave equation it is possible to control the damping of the sounds. The exponential damping factor $d = f\pi\tan(\phi)$ depends on the frequency f and internal friction ϕ of the material and causes higher frequencies to decay more rapidly. The internal friction parameter is material dependent and approximately invariant over object shape. Equation 3.2 represents the impulse response of a

general object at a particular point as a sum of damped sinusoids.

$$y(t) = \sum_{n=1}^{N} a_n e^{-d_n t} \sin(\omega_n t) \tag{3.2}$$

The sound model of an object consists of a list of amplitudes a_n and complex frequencies $\Omega_n = \omega_n + id_n$. Equation 3.3 shows how one complex frequency is computed for discrete time t. At time 0 the signal is the product of the frequency-amplitude a_n, and the contact force $F(0)$. At each successive time step (determined by the sampling frequency SR) the signal is the sum of a decayed and modulated version of the previous signal plus a new product of amplitude and contact force. The model responds linearly to input force $F(t)$. Once we have the model parameters, all we need to begin synthesizing sounds is a series of contact forces to plug into the right-hand side of the recursion. The output signal at time t is $\Im\mathrm{m}(\sum y_n(t))$, with the sum taken over all computed frequencies.

$$\begin{aligned} y_n(0) &= a_n F(0) \\ y_n(t) &= e^{i\frac{\Omega_n}{SR}} y_n(t-1) + a_n F(t) \end{aligned} \tag{3.3}$$

The linearity of the synthesis algorithm yields two benefits: it maintains the temporal information in the input signal, and it is amenable to an efficient anytime implementation. Since the computed audio is the discrete convolution of the force history with the impulse response there is a close relationship between temporal content of the input forces and the output signal. A "rough" input signal will generate a "rough" sound, a "smooth" input signal will create a "smooth" sound. This relationship allows for a tight coupling of continuous haptic and auditory feedback. If a single haptic contact triggered a recorded sample of a struck object it would suf-

fice for rendering contact events, but this technique would fail to generate scraping and sliding sounds.

The linearity of the synthesis algorithm also makes it efficient. With a basis change, an audio signal can be computed with 2 multiplications and 3 additions per mode per sample. If changing CPU loads leaves the audio interrupt routine with a variable amount of time to compute an audio signal it is possible to have the synthesis algorithm respond in an anytime fashion. After each complex frequency is computed we can check the amount of time remaining in our time slice. If it is less than the estimated time required to compute the next complex frequency, we exit the audio synthesis loop and return the current signal. By starting the computation at the lowest frequency and progressing upward, an increase in CPU load will result in a decrease in high frequency detail. This is preferable to having the audio "drop-out" or distort because audio samples cannot be computed quickly enough.

3.3.2 Gammatone Impulse Response Model

Equation 3.4 represents a general Gammatone impulse response. It is the same as Equation 3.2 but with an extra term of $t^{\gamma-1}$. Gammatone filters are used in cochlear models of auditory perception and are amenable to efficient synthesis [Slaney, 1993].

$$y(t) = \sum_{n=1}^{N} a_n t^{\gamma-1} e^{-d_n t} \sin(\omega_n t) \tag{3.4}$$

Incorporating Gammatone filters with the AHI gives us some finer control over the frequency content of our impulse response without sacrificing the benefit of time-invariant convolution and without an exorbitant cost. Figures 3.2 through 3.5 show the time domain and frequency domain responses of Gammatone filters for

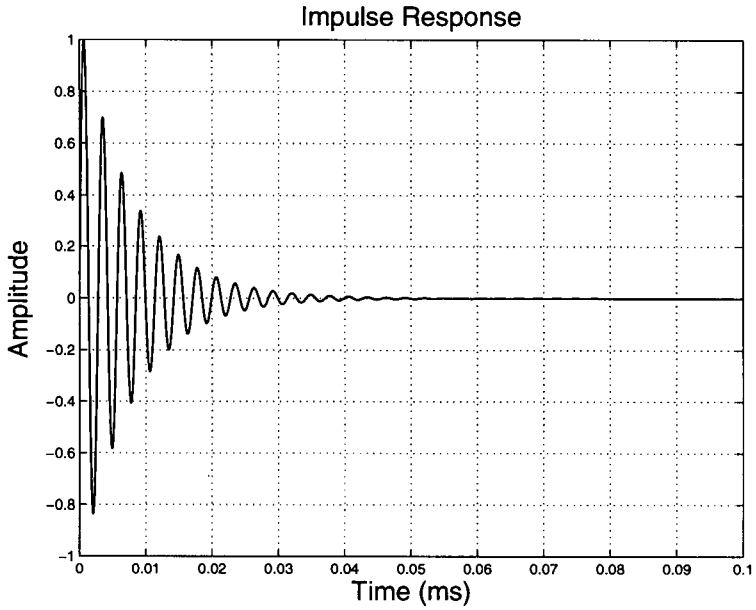

Figure 3.2: Time domain impulse response. The response peaks at $t = 0$.

$\gamma = 1$ (the trivial case) and for $\gamma = 2$. In the time domain $\gamma = 2$ responses have a gradual onset. In the frequency domain $\gamma = 2$ responses resonate more strongly at the peak frequency. Given this stronger resonance, and following Hermes, we use Gammatones with $\gamma = 2$ for synthesizing rolling sounds based on parameters from a rigid body dynamic simulation. We now derive Gammatone filters starting from the recursive form of Equation 3.3.

Equation 3.3 is a first order matrix equation for updating the complex variable y. Solving for $\mathfrak{Im}(y)$ gives an equivalent second order difference equation that can also be used for synthesizing audio.

$$y(t) = bF(t-1) + a_1 y(t-1) - a_2 y(t-2) \tag{3.5}$$

We replace $F(t-1)$ by $F(t)$ to keep the input signal in phase with the output signal (since the output y is no longer complex). The resulting second order difference

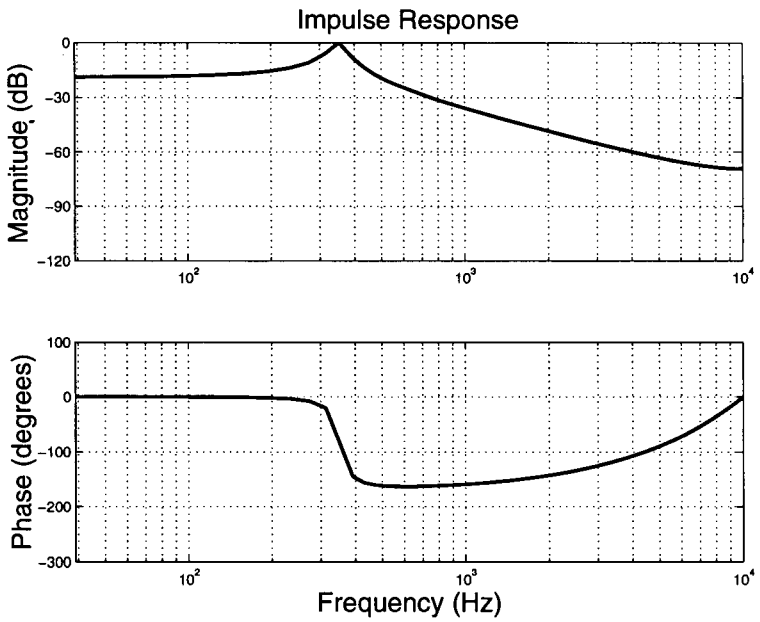

Figure 3.3: Frequency domain impulse response. In this example, $\omega = 350$ and $d = 40$.

using $F(t)$ as input is a two-pole IIR filter with a transfer function $H(z) = b/[1 - a_1 z^{-1} + a_2 z^{-2}]$. A Gammatone filter has a time domain representation as $ty(t)$. Consulting a table of z-transforms, the resulting transfer function is

$$-zdH(z)/dz = H(z)\left(\frac{-a_1 z^{-1} + 2a_2 z^{-2}}{1 - a_1 z^{-1} + a_2 z^{-2}}\right) \tag{3.6}$$

Multiplication in the frequency domain is equivalent to convolution in the time domain. A Gammatone filter can be synthesized from an impulse response by applying a second stage of convolution.

$$g(t) = a_1(g(t-1) - y(t-1)) - a_2(g(t-2) - y(t-2) - y(t-2)) \tag{3.7}$$

Equation 3.7 shows the extra filtering required. The variables g are local variables for this stage, and the y are input from Equation 3.5. The output signal is $g(t)$.

Computing Equation 3.7 as is requires an extra 2 multiplications and 4 additions per mode per sample. One subtraction can be saved by storing and reusing $(g(t-1) - y(t-1))$. An extra set of filter states $g(t), g(t-1)$, $g(t-2)$, and $(g(t-1) - y(t-1))$ need to be stored, but no new filter coefficients are required. This savings could be significant if the frequency or damping coefficients need to be scaled as a function of contact location. No special filter design considerations are required for this derivation other than the z-transform of Equation 3.6.

3.3.3 Synthesis of Rolling and Scraping sounds

To synthesize rolling and scraping sounds we would like to generate appropriate force input as a function of variables from a rigid body dynamic simulation. The model we describe here assumes that as the object rolls or scrapes small deviations in the surface profile create brief force impulses. These force impulses can be represented as a stream of arrivals with an exponentially distributed mean interarrival time of

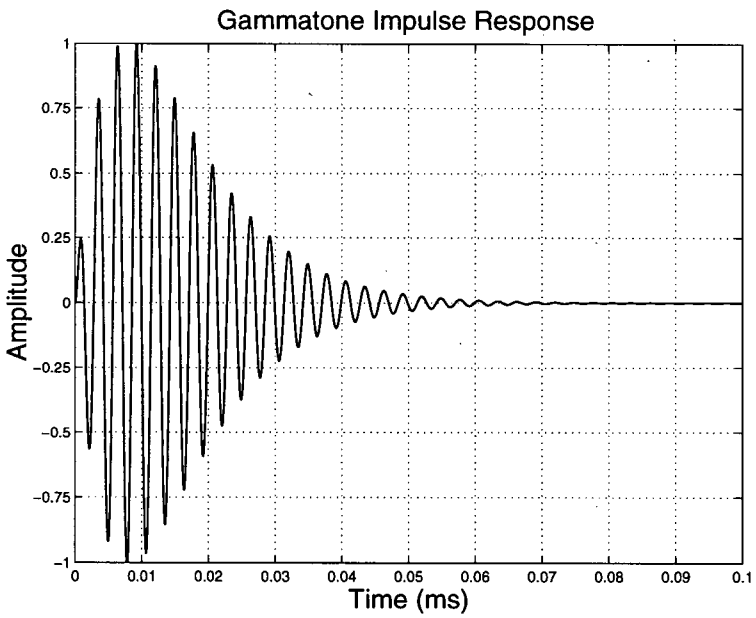

Figure 3.4: Time domain Gammatone impulse response. Note the slower onset of the output signal compared to the time domain impulse response in Figure 3.2.

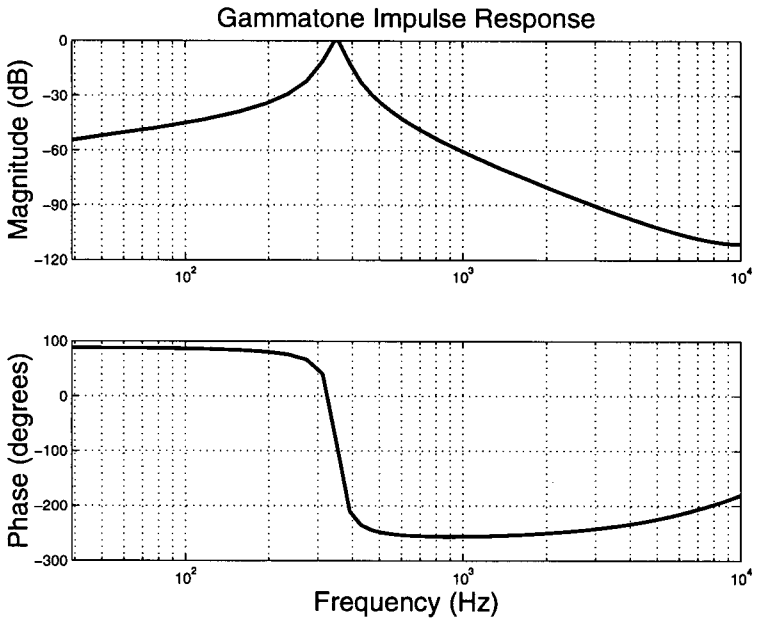

Figure 3.5: Frequency domain Gammatone impulse response. In this example, $\omega = 350$ and $d = 40$. Note the sharper resonance at 350Hz compared to the frequency domain impulse response in Figure 3.3.

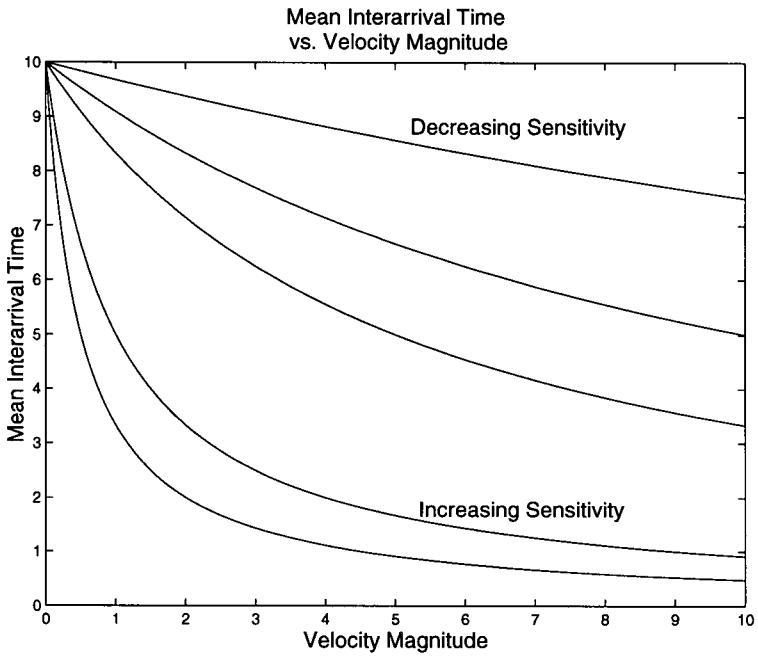

Figure 3.6: Mean interarrival time for impulses as a function of velocity magnitude. As the velocity magnitude increases, the mean interarrival time decreases. The sensitivity of the interarrival time to velocity magnitude can be controlled as well.

λ. Hermes only considers constant interarrival times λ [Hermes, 1998]. By mapping velocity magnitudes to an exponential distribution of interarrival times, we can vary the mean as the dynamic simulation changes state. The variables we consider are linear velocity magnitude $|v|$ for scraping and angular velocity magnitude $|\alpha|$ for rolling.

Figure 3.6 shows the particular mapping we use to transform velocity magnitudes to a mean interarrival time. The mapping function is

$$\lambda = \frac{p_{mean}}{1 + p_{sens}|v|} \tag{3.8}$$

where p_{mean} is the mean interarrival time λ at $|v| = 0$, and where p_{sens} controls the sensitivity of λ to increasing $|v|$. The motivation for this mapping is to allow

for exploration of different force profiles by interactively changing p_{mean} and p_{sens} during the dynamic simulation. We maintain two separate Poisson means, $\lambda_{|v|}$ for scraping forces as a function of linear velocity $|v|$, and $\lambda_{|\alpha|}$ for rolling forces as a function of angular velocity $|\alpha|$.

When a scraping impulse arrives the resulting force is $\log(1+|v|)$. This logarithmic amplitude modulation models the relation of input force to sound pressure level – the signal decays rapidly as $|v|$ approaches zero and increases more slowly as $|v|$ increases. Rolling impulses are also amplitude modulated by $\log(1+|\alpha|)$, but with the addition of a sinusoidal term as shown in Equation 3.9.

$$Force_{roll} = \log(1+|\alpha|)[1 + a_{depth}\sin(2\pi f_{roll}|\alpha|t)] \tag{3.9}$$

The depth of the sinusoidal modulation can be controlled by $a_{depth} \in [0,1]$ and the frequency of the modulation can be scaled by f_{roll}. Again, our motivation is to allow for interactive parameter exploration.

3.4 Haptic Force Synthesis

As the user moves the Pantograph handle we need to compute the contact forces resulting from these interactions and then render them as forces on the handle of the Pantograph by exerting torques on its base joints. These computations take place in two coordinate frames. One is the world frame of xy-coordinates and the other is the Pantograph frame of joint angles. The simulated environment uses the world frame but the control code only knows about joint angles. We need a forward kinematic mapping that gives the xy-position of the handle as a function of base joint angles as well as a differential kinematic mapping that gives the base joint torques as a function of applied force to the handle.

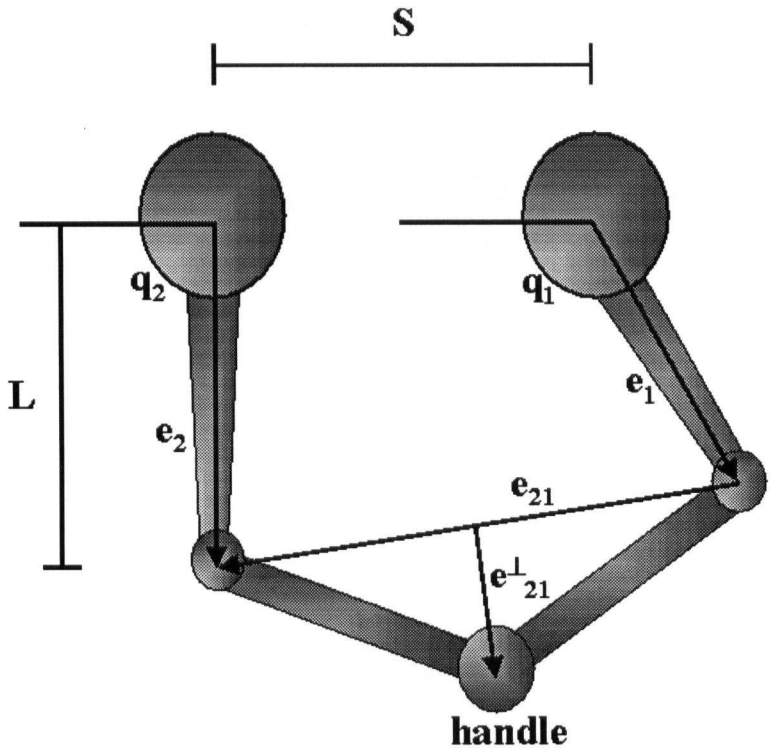

Figure 3.7: Pantograph kinematics. There is a geometric constraint that allows us to compute the position of the handle: the vector e_{21} pointing from elbow 1 to elbow 2 is always perpendicular to the vector $e_{21}^{\perp}$ pointing to the handle from the midpoint of e_{21}.

For the forward kinematic mapping we specify the base joint of motor 1 as the origin of the world frame. There is a geometric constraint that allows us to compute the position of the handle: the vector pointing from elbow 1 to elbow 2 ($e_{21} = e_2 - e_1$) in the world frame is always perpendicular to the vector pointing to the handle from the midpoint of e_{21}. If q_1 and q_2 are the base joint angles, then the elbows become $e_1 = (L\cos(q_1), L\sin(q_1))$ and $e_2 = (L\cos(q_2) + S, L\sin(q_2))$ where L is the length of the proximal arms and S is the separation between the two base joints. Setting $e_{21}^{\perp}$ as the vector pointing from the midpoint of e_{21} to the handle h,

we have $h(q) = e_1 + 0.5e_{21} + e_{21}^{\perp}$. This expression for h in terms of joint angles q has a simple geometric interpretation, as shown in Figure 3.7.

Once we have the handle coordinates, and compute a contact force F, we need to transform this force into base joint torques τ for rendering. The Jacobian $J = \partial h(q)/\partial q$ of the forward kinematic mapping relates forces to torques by $J^T F = \tau$. The details of constructing the Jacobian for the Pantograph are quite general and are covered in basic robotics texts [Murray et al., 1994]. In our particular implementation we can avoid the expense of computing the partials of $h(q)$ by exploiting the structure of the Jacobian.

3.4.1 Normal Forces

For interactions normal to the surface of a flat plane a spring/damper/impulse combination constrains the user to the surface by applying a penalty force. If the normal displacement past the surface is x_n, and the current normal velocity is v_n then the haptic constraint force is $F = Kx_n + Dv_n$ where K and D are spring and damping constants. For 10ms after a new contact we add a unilateral impulse Pv_n to the spring/damper combination; see Equation 3.10. This simple technique is known to increase the perception of haptic stiffness without introducing closed-loop instabilities that can occur with large spring coefficients [Salcudean and Vlaar, 1997].

$$F = \begin{cases} Kx_n + (D+P)v_n & \text{if } t <= 10ms \\ Kx_n + Dv_n & \text{if } t > 10ms \end{cases} \tag{3.10}$$

3.4.2 Tangential Forces

For interactions tangential to the surface of a flat plane we have implemented three friction models to provide force feedback: a viscous friction model and two stick-slip models.

Viscous friction force opposes tangential velocity,

$$f_{visc} = -d_{visc} v_t \tag{3.11}$$

where d_{visc} is a constant scaling factor.

The other two friction models are stick-slip models that exhibit two regimes: *slipping* and *sticking*. Specifying a stick-slip model requires defining transition conditions for moving between these two regimes [Armstrong et al., 1994]. In general, a virtual proxy point connects the real contact point to the surface. In the sticking regime the real contact point separates from the proxy and frictional force opposes further motion proportional to the separation. When a contact point is in a sliding regime the virtual proxy slides along with it.

Salcudean's stick-slip model uses a force maximum condition to transition from sticking to slipping and a velocity minimum condition to transition from slipping to stuck [Salcudean and Vlaar, 1997]. While in the stuck state the tangential force opposing motion is $f_s = k_s(x_{proxy} - x_t)$; while slipping, $f_s = 0$. Both x_{proxy} and x_t are tangential displacements. The state transitions from sticking to slipping when $|k_s(x_{proxy} - x_t)| > f_{max}$, and from slipping to sticking when $|\dot{x}| < v_{min}$. Immediately after any state transition, set $x_{proxy} = x_t$.

Hayward's stick-slip model only uses displacements to determine state transitions [Hayward and Armstrong, 2000]. If we define $z = x_k - x_{proxy}$ as the displacement between the real contact point and the proxy and z_{max} as the maximum

displacement then the update for the next proxy point is

$$x_{proxy} = \begin{cases} x_k \pm z_{max} & \text{if } \alpha(z)|z| > 1 \text{ (slipping)}, \\ x_{proxy} + |x_k - x_{k-1}|\alpha(z)z & \text{otherwise (sticking)}. \end{cases} \tag{3.12}$$

Once the displacement between the proxy and real contact point passes a maximum the contact becomes fully tense and enters the slipping regime. The proxy point and the real contact point move together, separated by z_{max}. For displacements less than the maximum the proxy point does not move much; this is the stuck regime. The non-linear adhesion map $\alpha(z)$ allows the proxy point to creep between these two regimes.

The algorithms and techniques we use for normal and tangential force generation are intended to be incorporated with more sophisticated applications that manage their own collision detection between complex polygonal geometries. For example, the Ghost API that comes with the PHANToM makes object creation and collision detection transparent to the user. Ghost computes normal forces using a version of Equation 3.10. In the next section we will see how these forces can be pre-filtered to produce coupled audio signals for the AHI.

3.5 Audio Force Synthesis

Naively using the raw normal forces produced by Equation 3.10 to synthesize audio yields poor results. There are three main properties of our synthesized haptic forces that can cause trouble. This section will describe how we remove these three properties from the haptic force by prefiltering. The filtered result is the *audio force* that we convolve with the stored impulse response in Equation 3.2.

The three properties of the haptic force that we wish to filter are as follows:

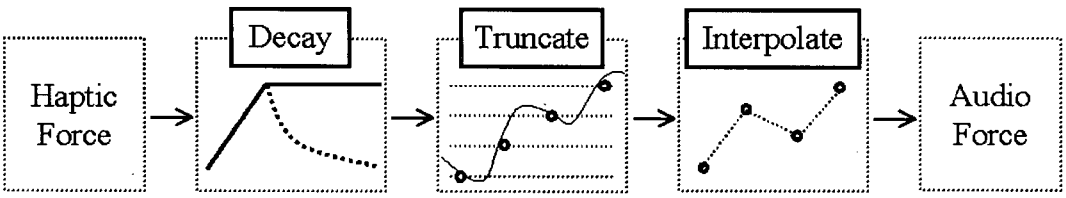

Figure 3.8: Prefiltering stages for haptic forces. Naively using the raw normal forces produced by Equation 3.10 to synthesize audio yields poor results. We need an intermediate stage of filtering before convolution with the audio impulse response. The filtered result is the *audio force* that we convolve with the audio impulse response.

(1) a spurious impulse that results when the user breaks contact with the surface and the haptic force discontinuously drops to zero, (2) high frequency position jitter, (3) control rate contamination. Any penalty method such as Equation 3.10 will always generate spurious audio impulses without prefiltering; however, the necessity of filtering the high frequency position jitter may only be a specific problem with our particular hardware. Linear interpolation can effectively remove the haptic control rate contamination. Figure 3.8 shows the three stages of haptic force prefiltering that remove spurious impulses, position jitter, and control rate contamination. Respectively, these stages are *decaying*, *truncating*, and *interpolating*. Each of these stages can be activated or deactivated as required.

3.5.1 Decay

Figure 3.9 plots an idealized haptic contact force. At 30ms the user comes into contact with the surface and stays in contact for another 30ms. Convolving this square wave profile with the impulse response of the surface will produce a spurious second "hit" when the user breaks contact. We introduce an attenuation constant β to allow the audio force to smoothly move to zero during sustained contact. If t

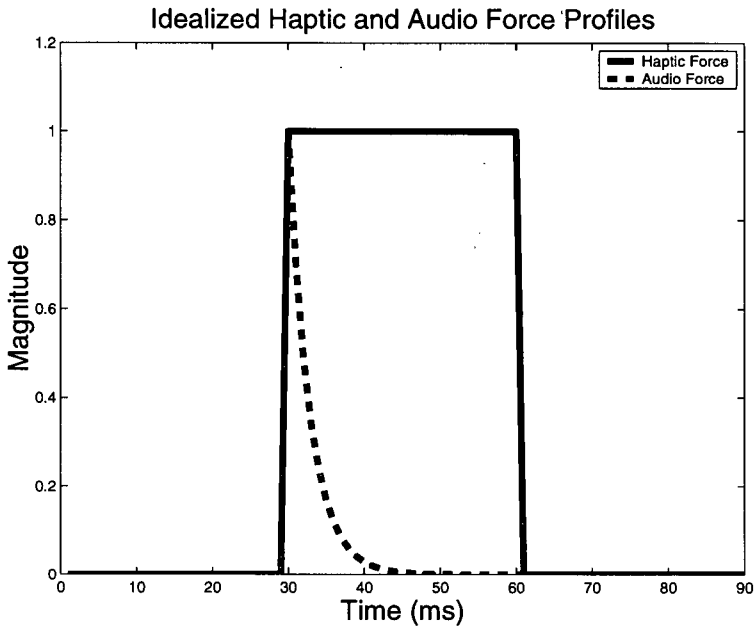

Figure 3.9: Idealized haptic and audio force profiles. The dashed line is an idealized example of a decayed haptic contact force with decay starting immediately after contact.

is the elapsed time since contact, then the current audio force is the current haptic force attenuated by β^t.

We have found that attenuating the audio force starting 10ms after a new contact with $\beta = 0.85$ (halflife of 5ms) produces good results. Waiting 10ms before decaying improves the percussive quality of impulsive contacts. Decaying the audio force immediately upon contact removes too much energy from the system and reduces the overall amplitude and dynamic range of the resulting audio signal. The dashed line in Figure 3.9 is an idealized example of a decayed haptic contact force with decay starting immediately after contact.

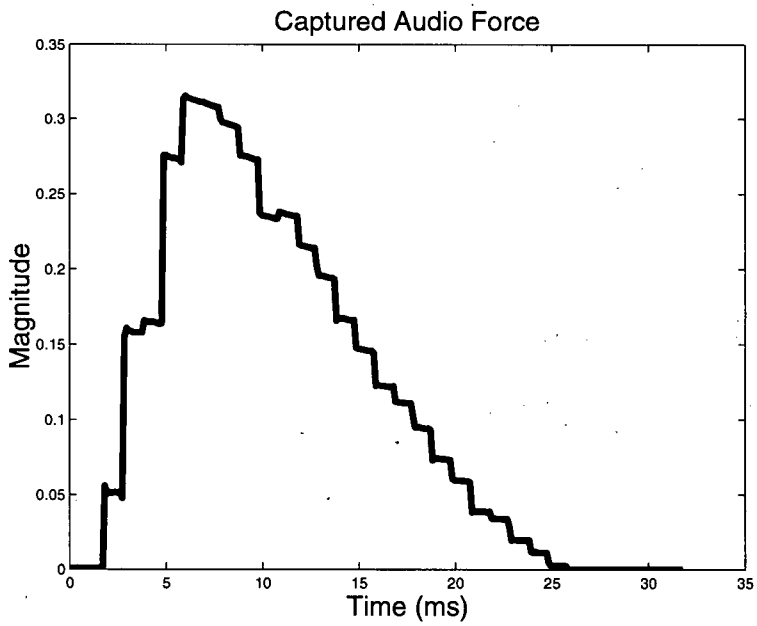

Figure 3.10: A synthesized audio force profile, captured as an input audio signal to a soundcard. The signal is constant over 1ms intervals (no interpolation), and the signal is decayed starting at 10ms.

3.5.2 Truncate

Haptic instabilities and signal noise generate sustained low amplitude, high frequency jitter. There is noise in the voltage readings from the potentiometers, as well as ADC conversion noise. These high frequencies are passed into the haptic force profile by the linear spring constant. Without filtering this noise becomes audible as crackling and popping while the user maintains static contact with the surface.

This low amplitude noise can be mostly eliminated by truncation. Typically, we remove the 8 lowest order bits. We chose truncation over averaging (low-pass) filtering because it was equally effective at a lower computational cost. Figure 3.10 plots a typical audio force profile. (The signal is not perfectly constant during each millisecond interval because it was captured as the input signal to a soundcard.) Using discrete optical encoders instead of potentiometers to read joint angles removes the need to truncate the position signals – the finite resolution of the encoders effectively truncates the signal for us.

3.5.3 Interpolate

The final stage of prefiltering is linear interpolation. As the decay time of any sound model goes to zero the resulting convolution approaches the original force profile. We can expect to hear more haptic control rate bleeding into the audio signal for rapidly decaying audio impulse models. If the haptic force signal is not interpolated, the resulting audio signal will approximate a 1kHz square wave. Linear interpolation removes the majority of this effect at a low computational cost, as well as the cost of a one sample delay (typically 1ms) between the output haptic signal and the audio signal.

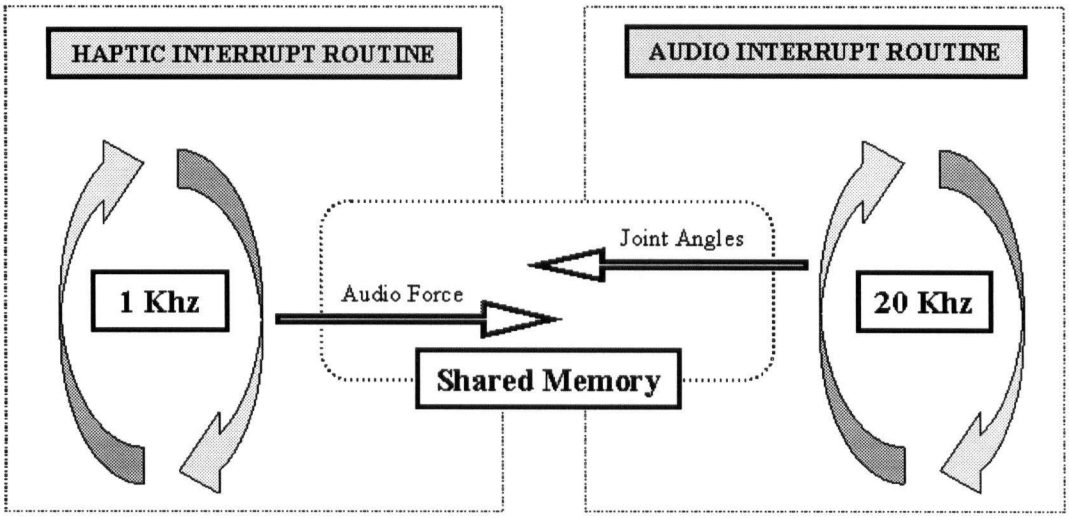

Figure 3.11: A high level view of real-time simulation with the AHI. The haptic interrupt routine computes contact interactions (typically at 1kHz) and sends audio forces to the audio interrupt routine for convolution (typically at 20kHz). The audio interrupt routine reads the pantograph joint angles and sends them to the haptic interrupt routine for mean filtering. The two interrupt routines communicate via shared memory.

3.6 Real-Time Simulation

The basic control structure for the AHI real-time synthesis and simulation is interrupt driven. There is a haptic interrupt service routine (HISR) that generates haptic and audio forces and an audio interrupt service routine (AISR) that convolves the audio force with the impulse response of the modeled object. Using these two separate interrupts we can synthesize the audio signal at a much higher rate than we generate haptic feedback. This section will describe the two interrupt routines shown in Figure 3.11.

3.6.1 Audio Interrupt Service Routine

The AISR and all DAC/ADC latches are synchronized to trigger at the audio update rate by using a programmable interval timer that counts at half the ISA bus clock rate of 8.33 MHz. The AISR reads the Pantograph joint angles from the ADCs and stores them in an array that contains a history of joint angle readings. Converting the DAC input to an equivalent floating point number requires 1 comparison, 2 multiplications and 2 additions. The current audio force is the sum of the filtered normal force and tangential force. The sum is then clipped to lie between 0.0 and 1.0 and truncated to remove low amplitude noise. This requires 2 comparisons and 2 multiplications. Interpolation requires 1 addition (to increment the previously stored value). A discrete convolution step using this filtered audio force $F(k)$ produces the output audio signal $y_n(k)$. This signal is placed in the DAC out. Computing the audio signal requires 3 multiplications and 3 additions per complex frequency.

If the DSP is short on cycles we can decrease the number of active frequencies. In our current scheme this isn't necessary – there are no other competing processes for DSP time. Once a number of complex frequencies are selected the total amount of processing time is fixed and does not need to be adaptively adjusted. This would change if the DSP was also managing a complicated environment with graphics, collision detection, and dynamics.

3.6.2 Haptic Interrupt Service Routine

Haptic interrupts trigger at an integer fraction of the audio update rate. We schedule these interrupts by using a timer that counts at the chip clock rate of 40 MHz. During this interrupt the current joint angles are computed as the mean of the array of joint angles that are read during the audio interrupt routine. This mean

filtering noticeably improves the stability and feel of the haptic walls and reduces noise in the resulting audio signal. From these filtered position values we use the forward kinematics of the Pantograph to compute the position of the handle. Since we only consider interactions with a plane, determining contact between the handle (represented as a point) and the plane takes a sign check. If there is contact we compute a normal force using Equation 3.10 and a tangential force using one of the friction models in Section 3.4.2.

To synthesize rolling and scraping force profiles we need to generate impulses with a exponentially distributed interarrival time. Given a current mean λ we compute the interarrival time as $p_{time} = \lceil -\lambda \log(U) \rceil$, where U is a uniform random number between 0 and 1. At each time step a counter variable, initialized to p_{time}, is decremented. Once the counter falls below zero, an impulse is triggered, and the counter is reset to a new value of p_{time} . Generating a new p_{mean} requires one call to a $rand()$ function, one multiply, and two comparisons and an addition for the ceiling operation. When the mean λ does not change we only need to decrement the counter variable. For scraping we need one logarithm call for each impulse; for rolling we need one logarithm call, one trigonometric call, three multiplications, and one addition. In our implementation Poisson streams for rolling forces are updated in the HISR and Poisson streams for scraping are updated in the AISR. Our experience is that scraping sounds require higher frequency Poisson streams than can be generated by the 1kHz HISR.

Computing the current Jacobian takes 22 multiplications, 4 trigonometric calls, and 2 square roots. The Jacobian of the Pantograph transforms the haptic force into motor torques. The voltages to generate these torques are written to the DACs. If there has been contact for $(10 + t)$ milliseconds, then β^t times the haptic

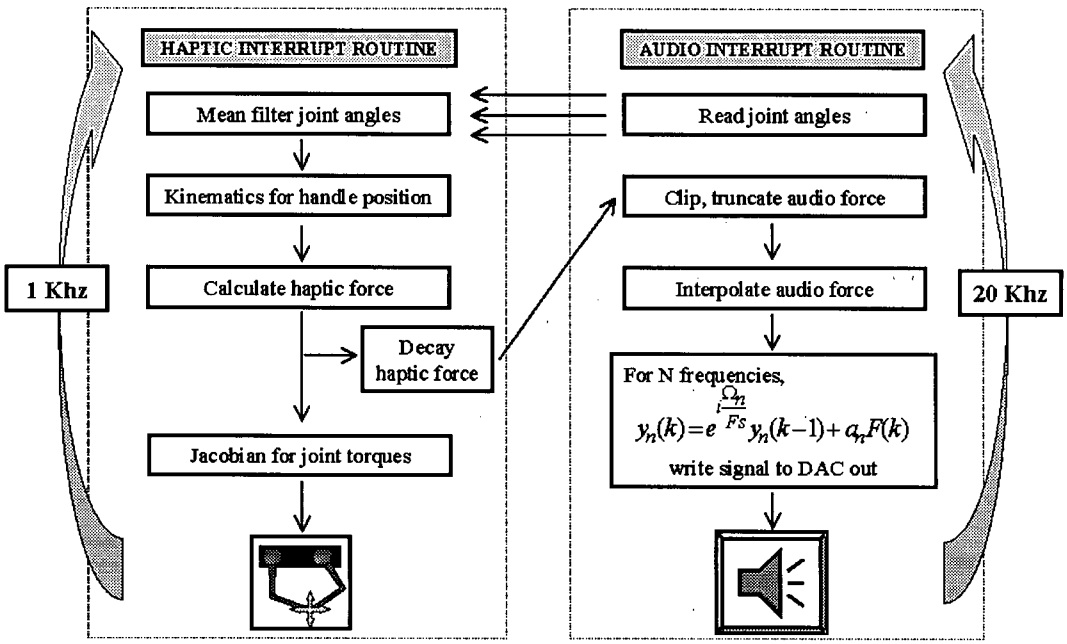

Figure 3.12: Flow of control for real-time audio and haptic simulation with the AHI. The control rates (1khZ HISR and 20kHz AISR) are typical values, but can be changed if necessary and if enough processing power remains. The two interrupt routines communicate via shared memory.

force becomes the current audio force. The HISR writes the current audio force to a global variable shared with the AISR. This global variable is only written to by the HISR and only read from by the AISR, so there is no need to implement a variable lock.

Figure 3.12 illustrates the flow of control for real-time audio and haptic simulation with the AHI. Typical control rates of 1kHz and 20kHz are shown, but these rates can be changed if necessary and if enough processing power remains. Communication between the two interrupt routines occurs via shared memory.

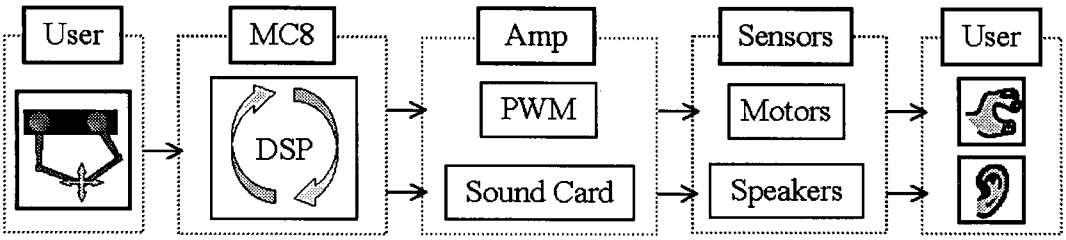

Figure 3.13: Characterizing the system latency for the AHI. We define end-to-end system latency as the time for system input, DSP, and system output. The sum of the analog and digital delays at each stage is small enough to assume that the haptic control rate on the DSP dominates the system latency. For a 1kHz HISR rate, this results in a nominal 1ms system latency. The maximum asynchrony between the output audio and haptic signals is equal to one AISR control period plus any difference in subsequent amplification and rendering of the two streams. An audio control rate of 20kHz has a period of $50\mu s$. Different conversion rates in the amplification stage result in an additional delay of the haptic signal by $45\mu s$. A conservative estimate for expected asynchrony is about 5% of the nominal system latency, but will vary if the amplification stages are changed.

3.7 Latency and Asynchrony

In general, we define end-to-end system *latency* as the time for system input, DSP, and system output. We also define *asynchrony* as the temporal separation between the output haptic signal and output audio signal. This section describes each step in the AHI feedback loop with the goal of estimating the AHI's expected latency and asynchrony.

3.7.1 Timing Resolutions

The programmable interval timer (PIT) on the MC8 determines our timing resolution for interrupt handling and for instruction execution. The PIT counts at half of the ISA bus clock rate. This results in a 240ns resolution, or about 0.24% of a nominal 1ms latency.

The audio control rate on the MC8 determines our timing resolution for when ADC/DAC conversions occur. A typical audio control rate of 20kHz results in a $50\mu s$ resolution between consecutive ADC/DAC conversions, about 5% of a nominal 1ms latency. The total latency and asynchrony between audio and haptic signals output by the MC8 will be an integer multiple of this resolution. Subsequent amplification and rendering will also contribute to latency and asynchrony.

3.7.2 Processing Haptic and Audio Streams

Figure 3.13 illustrates the end-to-end processing steps between receiving position input and returning haptic and audio signals. The first step is to read the potentiometer voltages and encoder counts. Conversion times for the ADC dominate this stage. The Precision Microdynamics manual states that ADC/DAC conversion completes by $10\mu s$ after the latching signal. The next step is the internal DSP for audio and haptic interrupts described in detail in the previous section.

DSP on the MC8

It is possible to have to wait for one full haptic interrupt before using the input position to compute force. This is typically 1ms. We have logged the amount of time spent in the HISR by comparing the PIT values at the start and end of the HISR. Table 3.1 shows typical HISR processing times. These values include the time lost while being interrupted by the AISR, and also include the interrupt overhead for context switching. The relevant values are when the user is in contact with the surface. When in contact, the HISR needs additional DSP cycles to compute contact forces (perhaps including tangential friction) and motor torques.

Once a contact force is computed, the haptic and audio streams diverge. The

	Not In Contact	In Contact
20 Modes	$62\mu s$	$100\mu s$
5 Modes	$29\mu s$	$48\mu s$

Table 3.1: Measured processing times for the HISR. These include the time lost while being interrupted by the AISR, and also include the interrupt overhead for context switching. The relevant values are when the user is in contact with the surface. When in contact, the HISR needs additional time to compute contact forces and motor torques.

current haptic force leaves the MC8 at the next latch time. However, this haptic force still needs to be convolved with the stored impulse response in the AISR. Regardless of the number of modes computed, the audio signal for the current haptic force will not leave the MC8 until the next latch time $50\mu s$ later. After DSP for both streams, there is another stage of DAC conversion, which adds another $10\mu s$.

Amplification

The Copley PWM amplifiers have a bandwidth of 3kHz, and perform voltage to current conversion at 22kHz. We assume that two conversions are necessary to read a voltage and write a current. Two conversions at 22kHz require $90\mu s$. The Soundblaster audio card adds another set of ADC/DAC conversion to the audio stream. Again, we assume that two conversions are necessary to read a voltage and write a current. Two samples at 44.1kHz require $45\mu s$. Using a linear analog amplifier for the audio stream would eliminate this $45\mu s$ sampling delay.

Sensors

The large Maxon motor's mechanical time constant is 5ms. Applying a step function, the motors will have reached about 20% of the full step value after 1ms (using $1 -$

$e^{-1/5} = 0.18$). We expect (but have not measured) that the resulting torque excites structural vibrations in the 5-bar mechanism that can be felt immediately when the step is applied. The mechanical time constant of the speakers (or headphone speakers) will be much lower than that of the motors.

Expected Latency and Asynchrony

We can sum the various contributions to latency and asynchrony for the haptic and audio streams. Figure 3.14 shows the relevant internal processing in Figure 3.13. For both the haptic and audio stream there is a $10\mu s$ ADC conversion, a 1ms haptic interrupt, approximately $100\mu s$ of processing time in the HISR (from Table 3.1). At this point the synchronized haptic and audio streams diverge. The computed haptic forces are converted at the next latch. This requires another $10\mu s$. However, the audio signal computed from this force must wait until the next conversion before leaving the MC8. The next conversion does not begin for another $50\mu s$ and then we must add the $10\mu s$ conversion time. Amplifying these streams adds more system latency, $90\mu s$ for haptics and $45\mu s$ for audio.

Expected system latency for the haptic stream is 1.20ms and for the audio channel is 1.205ms when computing 20 modes. Expected asynchrony is the difference between these two numbers: $5\mu s$, about 0.5% of the nominal haptic control rate of 1ms. This number is too close to the timing resolution to be trustworthy. A more conservative estimate for expected asynchrony is $50\mu s$, based on the one AISR control period delay to update the audio signal. We have not measured these end-to-end values or asynchronies. Our only empirically measured value is the amount of time spent in the HISR. We will assume that the HISR control rate determines the end-to-end system latency, and will define the nominal system latency as equal

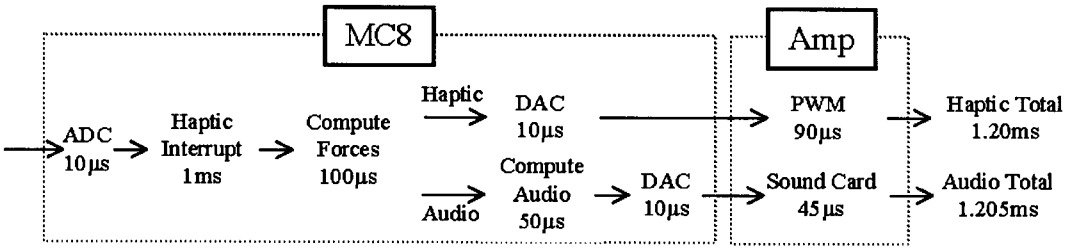

Figure 3.14: Summing the contributions to system latency and asynchrony. For both the haptic and audio stream there is a $10\mu s$ ADC latch, a 1ms haptic interrupt, and approximately $100\mu s$ of processing time in the HISR (from Table 3.1). At this point the synchronized haptic and audio streams diverge. The computed haptic forces are converted at the next latch. This requires another $10\mu s$. The audio signal computed from this force must wait until the next conversion before leaving the MC8. The next conversion does not begin for another $50\mu s$ and then we must add the $10\mu s$ conversion time. Amplifying these streams adds more system latency, $90\mu s$ for haptics and $45\mu s$ for audio. In total, expected system latency for the haptic stream is 1.20ms and for the audio channel is 1.205ms when computing 20 modes.

to the period of the HISR.

3.7.3 Interrupt Priorities and Asynchrony

The DSP chip allows us to prioritize the two interrupts when they are first scheduled. A higher priority always interrupts a lower priority interrupt and conversely a lower priority never interrupts a higher priority interrupt. It is necessary to schedule the AISR as higher priority than the HISR to ensure that audio samples are not dropped. If the scheduling is reversed then the resulting audio signal will be corrupted by a waveform at the haptic update rate (typically 1 or 2 kHz). The resulting corruption is audible, and not pleasing.

This cost of this interrupt priority convention can be starvation of the HISR for processor time. As the amount of time spent in the AISR increases, the amount of time required to compute a new haptic force will also increase. End-to-end la-

tency will increase accordingly, in multiples of the audio control period. However, asynchrony between the haptic and audio stream will not change on the MC8. Regardless of when the haptic force arrives, there will always only be a single AISR control period delay between the output haptic force and the corresponding audio signal on the MC8.

3.8 Chapter Summary

This chapter has presented the system details of the AHI. Our implementation combines haptics and audio synthesis algorithms with dedicated hardware support. Both haptic and auditory stimuli are produced from the same surface model for contact interactions. This choice of representation, coupled with dedicated hardware, allows the AHI to synthesize auditory and haptic stimuli with an end-to-end system latency of 1ms. A conservative estimate for expected asynchrony between the two stimuli is on the order of $50\mu s$, or 5% of the nominal system latency. The next chapter will present our evaluation of the usefulness of the AHI for conducting perceptual studies, the feedback we received from presenting the AHI as a live demonstration, and some preliminary results obtained by integrating the AHI with an interactive dynamic simulation.

Chapter 4

Evaluation and Results

4.1 Overview

The previous chapter described real-time audio and haptic simulation with the AHI. By using specialized hardware and efficient algorithms we are able to reduce our total system latency for synchronized auditory and haptic feedback to 1ms. There are no other human-computer interfaces we know of that can produce the kind of continuous and synchronized stimuli rendered by the AHI.

As a result of its novelty there are no established performance or perceptual benchmarks with which to objectively evaluate the quality of the AHI. After showing some typical examples of the kind of contact interactions made possible by the AHI we will describe the results of an informal user study which we conducted to help us verify that our system latency was below the perceptual threshold for detecting simultaneity. Our main goal was to evaluate the suitability of the AHI for conducting more thorough user studies in the future. The AHI was presented as a live demo for three consecutive days at the Institute for Robotics and Intelligent Systems (IRIS) conference in May 2000, in Montreal. We will describe the Java GUI front-end

Figure 4.1: A brass vase, 10cm tall. The ACME facility measured its audio impulse response at a point about 3cm from the base. This impulse response was used by the AHI for rendering sounds. Note the wavy vertical surface profile around the lower half of the vase.

that users could interact with and recount some of the constructive criticism and positive feedback we received. Finally, we will describe some preliminary efforts towards integrating the AHI with a physically-based rigid body dynamic simulation also implemented in Java.

4.2 Contact Interactions

We have experimented with some examples using the basic control structure described in Figure 3.12 to demonstrate how the AHI can generate continuous audio and haptic interactions. In the following examples the user taps and scrapes the AHI handle across surfaces with different stiffness properties. We convolve the resulting

Figure 4.2: A softer force profile. The impulse constant of the surface was set to zero, the spring constant was reduced, and normal force attenuation was disabled. Interpolation was enabled.

audio force with the impulse response of a brass vase acquired using the ACME facility. Figure 4.1 shows the actual vase. We compute 20 modes for each audio sample. In the following examples, haptic interrupts trigger at 1kHz and audio interrupts at 20kHz. There is a 1ms latency for changes in force and audio, unless interpolation is enabled. In this case, the audio signal lags the haptic signal by 1ms. Informally, the auditory and haptic stimuli are perceptually simultaneous.

Figures 4.2 and 4.3 show a captured force profile and a 256-point spectrogram of the resulting audio signal after convolution. In this example the impulse constant of the surface was set to 0, the spring constant was reduced, and normal force attenuation was disabled. The aim was to simulate a soft strike like hitting the vase with a soft mallet. The interaction consisted of a single strike normal to the flat

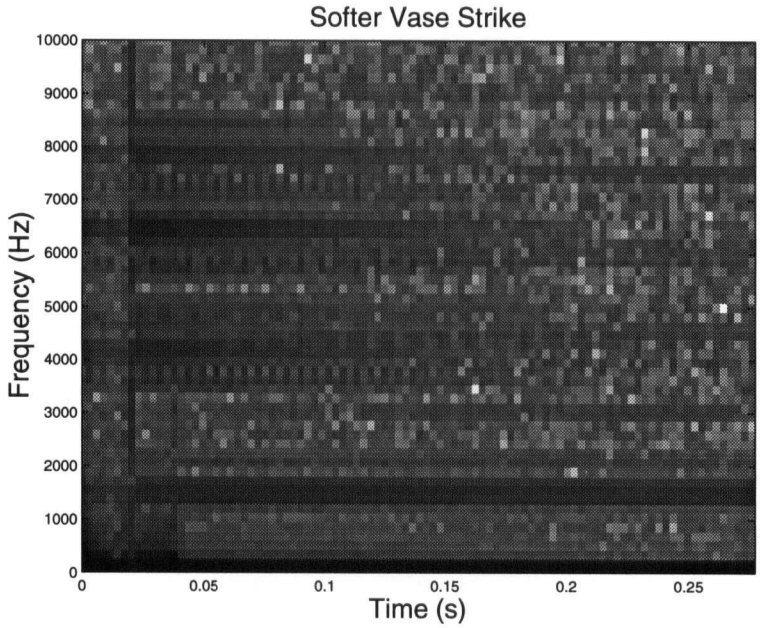

Figure 4.3: Spectrogram of a soft vase strike after convolution with the force profile in Figure 4.2.

surface.

Figures 4.4 and 4.5 also show a captured force profile and a 256-point spectrogram of the resulting audio signal after convolution. In this example the impulse constant and the spring constant of the surface were increased, and the normal force attenuation was still disabled. The aim was to simulate a harder strike. In Figure 4.4 the initial velocity impulse dominates the spring force for the first 10ms. The corresponding broadband energy in the resulting spectrogram reflects this initial force impulse.

One must be careful not to let the haptic control rate corrupt the audio signal. The spectrograms in Figures 4.3 and 4.5 are not corrupted by the 1kHz haptic interrupt rate because, for these examples, we linearly interpolated the HISR

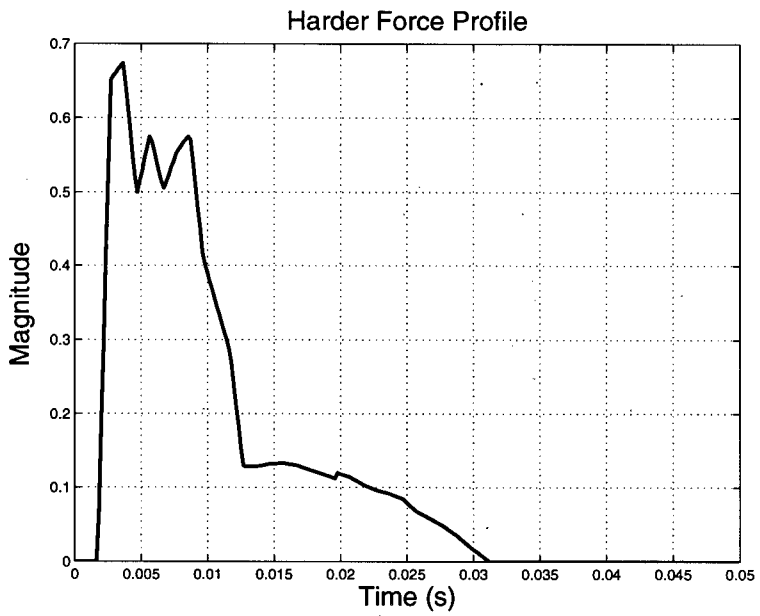

Figure 4.4: A harder force profile. The impulse constant and the spring constant of the surface were increased and the normal force attenuation was disabled. Interpolation was enabled. The initial impulse dominates the spring force for the first 10ms.

forces in the AISR. The dominant mode for the vase model occurs at about 1500Hz and can be clearly seen in the spectrogram. The faint bands at 3kHz, 4.5kHz, etc., are aliased versions of the dominant mode of the vase model. Figure 4.6 shows a 256-point spectrogram of a force profile that is *not* linearly interpolated. The aliased harmonics at multiples of the haptic control rate are clearly visible. Nevertheless, convolving this force profile with the vase impulse response still sounds like a vase and not a 1kHz square wave. Without interpolation, a quiet whine can be heard imposed on the sound of the vase. Linear interpolation removes this whine at very little computational cost, but also at the cost of having the audio signal lag the haptic signal by one force sample.

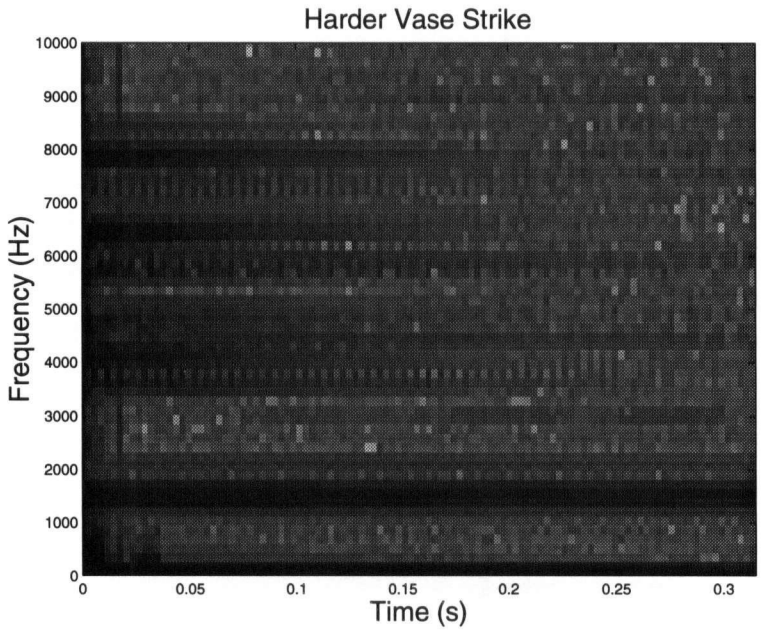

Figure 4.5: Spectrogram of a hard vase strike after convolution with the force profile in Figure 4.4. The broadband energy in the first 10ms results from the velocity impulse.

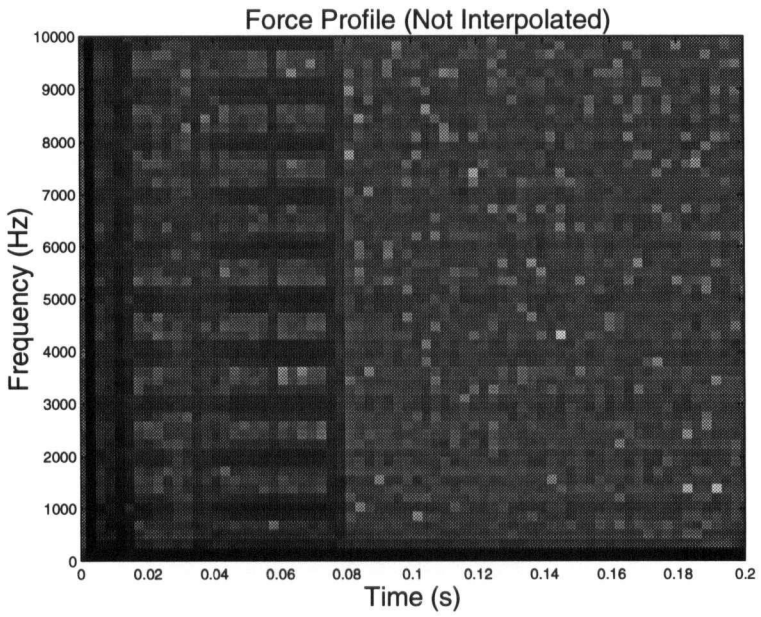

Figure 4.6: Spectrogram of a force profile without linear interpolation of HISR forces in the AISR. The harmonics at multiples of the 1kHz haptic control rate are clearly visible. Convolving this force profile with the vase model still sounds like a vase, however a quiet 1kHz whine can be heard for extended quasi-static interactions.

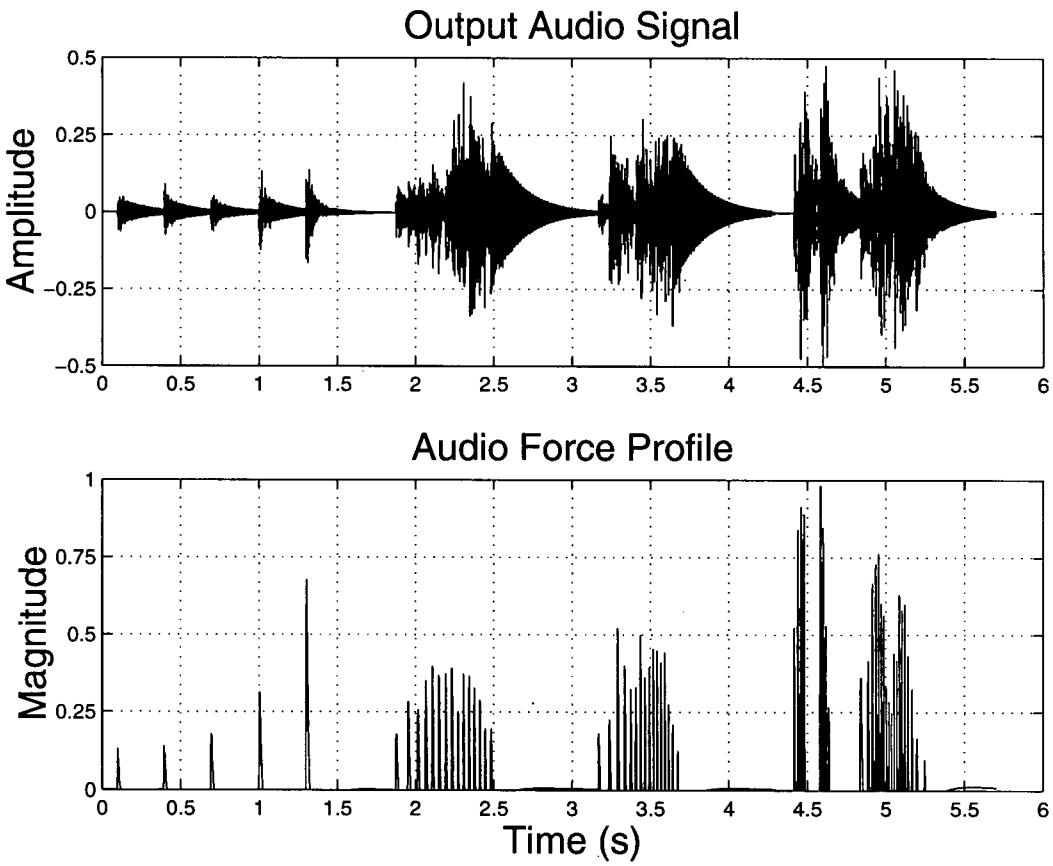

Figure 4.7: Recorded signals from interacting with a model of the brass vase from Figure 4.1. The user interaction in this example consisted of five single strikes of increasing force normal to the surface, then tangential motion across the surface. Force interpolation was disabled. The lower plot shows the audio force that is convolved with the vase audio impulse response. The upper plot shows the resulting output audio signal.

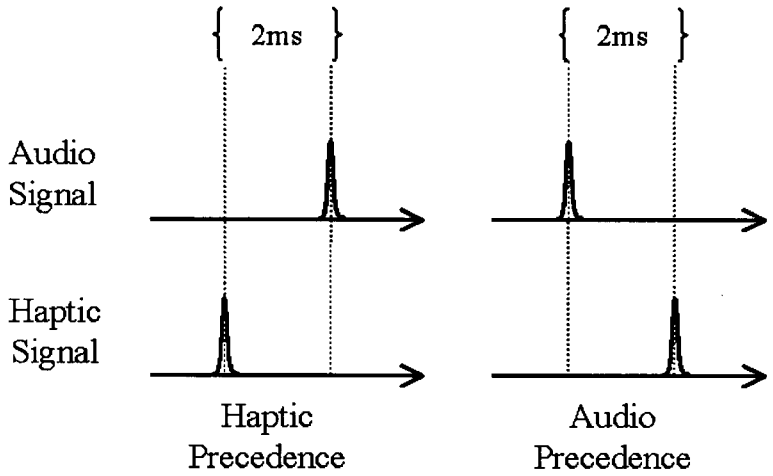

Figure 4.8: Audio and haptic precedence of 2ms.

In the next example, the user scrapes the AHI handle across a sinusoidally modulated surface profile. Figure 4.7 plots the captured audio force and the convolution of this force with the measured impulse response of the vase. The user interaction in this example consisted of five single strikes of increasing force normal to the surface, then tangential motion across the surface. The middle two bursts are slower scrapes back and forth, and the final two bursts are faster scrapes. The auditory signals produced in this fashion are satisfying. "Zipper" audio effects can be created by rapidly scraping on the surface. These synthesized audio signals compared favorably to live signals of tapping and scraping along the ribbed surface of the vase in Figure 4.1 with the tip of a pen.

4.3 User Study

This section will describe a pilot user study we conducted with the AHI. In this user study we tested the hypothesis that a 2ms asynchrony between auditory and haptic

events lies below the perceptual tolerance for detecting synchronization between auditory and haptic contact events. We selected 2ms because it is larger than our expected system asynchrony but a small enough interval to establish a lower bound. Briefly, a subject tapped on a virtual wall and received appropriate auditory and haptic stimuli except that one of the two was delayed by 2ms. We tested the hypothesis that all subjects would perform at chance when asked to choose which stimulus came first. Figure 4.8 plots two idealized audio and haptic signals. In the first column, the haptic signal leads the audio signal by 2ms. We call this haptic precedence. In the second column, the audio signal leads by 2ms and this is audio precedence.

4.3.1 Participants

Twelve members of our department (2 females, 10 males) participated in the user study. Their mean age was 32 years with a minimum age of 21 and a maximum age of 55. There was one left-handed participant. All twelve reported normal hearing and touch. The participants were not paid for their time.

4.3.2 Apparatus and Stimuli

With a few software modifications to the control code and Java GUI, we configured the AHI for this user study. The stimuli consisted of 24 contact events where the audio signal preceded the haptic by 2ms, and 24 contact events where the haptic signal preceded the audio signal by 2ms. The haptic control rate was 2kHz and the audio control rate was 20kHz, resulting in a nominal 0.5ms system latency. We created precedence by delaying one of the two signals with a short circular buffer. Delaying the haptic signal did not cause any instability. We disabled tangential

forces.

A vertical "wall" was positioned in the right half of the workspace. One set of 48 random locations of the wall within ±1cm were generated and used in the same order for all subjects. Haptic force-feedback was provided using the spring/damper/impulse combination in Equation 3.10. The instantaneous impulse is very important for creating sharp auditory and haptic signals. If the impulse is removed both signals become less crisp and as a result less suitable for judgments of simultaneity.

For audio synthesis, striking the wall was treated as striking a single point on an ideal bar clamped at both ends. The contact point was at 0.61 of the length of the bar. For a given fundamental frequency this simple geometry has an analytical solution for the frequencies and relative amplitudes of the higher partials [Morse, 1968]. We used a total of 5 damped sinusoids with fundamental frequencies ω_1 of 1000Hz (High) and 316Hz (Low) and four higher partials.

As shown in Equation 3.2, the decay of these frequencies are determined by a damping coefficient $d = f\pi\tan(\phi)$. We used two values of $\tau_d = 1/(\pi\tan(\phi))$ that correspond to slow decay and fast decay: 300 (Fast), and 3 (Slow). These particular auditory stimuli were selected because they are a subset of those used for a study that connects variation of the coefficient τ_d to auditory material perception [Klatzky et al., 2000].

4.3.3 Experimental Design

The experiment used a two-alternative forced choice design. The subjects were asked to decide whether the audio signal preceded the haptic signal or the haptic signal preceded the audio signal.

Audio/Haptic Precedence	Frequency	Damping
Audio	High	Fast
Audio	High	Slow
Audio	Low	Fast
Audio	Low	Slow
Haptic	High	Fast
Haptic	High	Slow
Haptic	Low	Fast
Haptic	Low	Slow

Table 4.1: Three factor within-subject design for the user study leads to eight different types of stimuli for the subjects: Audio or Haptic precedence, High or Low frequency, and Fast or Slow damping.

A three factor within-subject design was used with audio/haptic precedence, frequency, and damping as the within-subject factors. In total there were 8 different stimuli presented to the subject. Audio precedence coupled with one of four frequency/damping combinations (High + Fast, Low + Fast, High + Slow, Low + Slow) and haptic precedence coupled with the same four sound combinations. Six of each of these 8 different types were permuted once and used in the same order for all 12 subjects for a total of 48 stimuli.

4.3.4 Experimental Procedure

Subjects sat on a chair with the Pantograph on a desk in front of them. The Pantograph base was affixed to the desktop with a rubber sheet to minimize sliding and rotating. Subjects wore closed headphones (AKG K-240) for the audio signals and to minimize external sounds including those from the Pantograph device. We told the subjects that they would be striking a virtual wall and that this would produce both haptic forces and audio signals. They were told that in each case one

of the stimuli preceded the other. We demonstrated how to hold the handle of the pantograph and how to make an impulsive strike to the wall. Then the subjects were allowed to practice a few strikes with the headphones on. Finally, the experiment began.

Subjects were not told that there were equal numbers of stimuli types nor were they told the number of repetitions in the experiment. No requirement on striking force was suggested – subjects were free to strike the wall as firmly or softly as they wished as long as it was a single strike. There were no visual cues, but the subjects were not blindfolded.

After being read the instructions (included in Appendix A) the subjects were allowed to ask questions about the purpose of the experiment. If they expressed some concerns about their ability to discriminate between the two alternatives they were told that the discrimination task was designed to be difficult and to expect some ambiguity. We stored all response data on a secure UNIX file system with all group and world read/write permissions disabled.

4.3.5 Results

Figure 4.9 displays the ratio of correct to incorrect responses by individual subject. The darker slices in this figure represent the proportion of correct responses. Most subjects performed just below chance. No subject performed significantly above chance, and only one subject (third row, second column) performed significantly below chance.

Figure 4.10 displays the ratio of correct to incorrect responses by factor. The darker slices in this figure represent the proportion of correct responses. There appears to be a bias towards selecting haptic precedence with fast damping (first

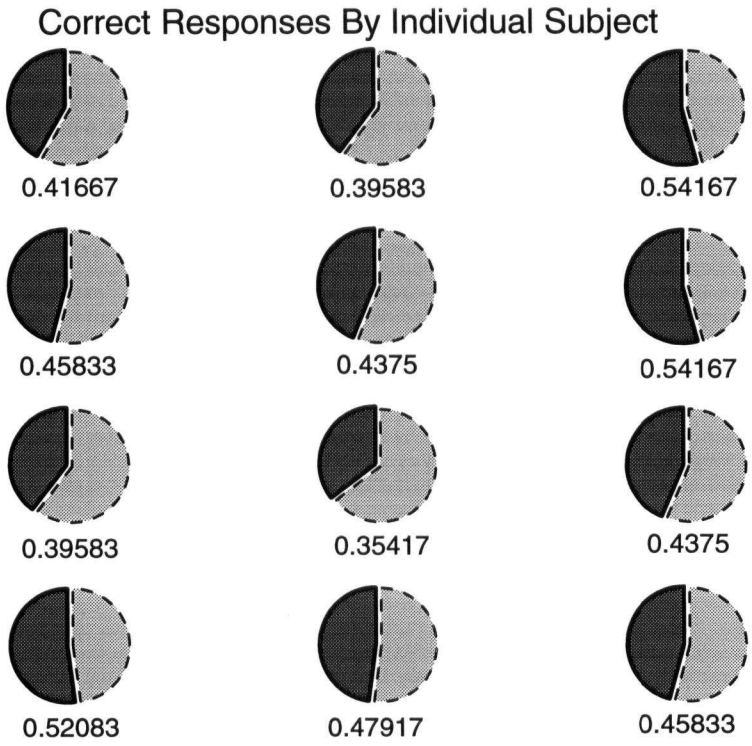

Figure 4.9: The ratio of correct to incorrect responses by individual subject. The darker shaded slices represent the proportion of correct responses. Most subjects performed just below chance.

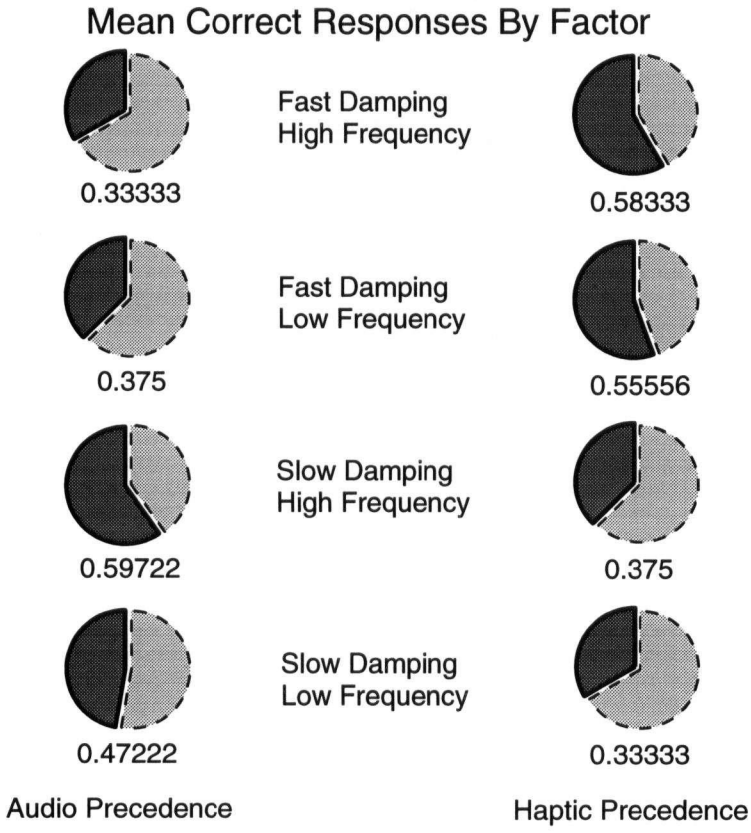

Figure 4.10: The ratio of correct to incorrect responses by factor averaged across all subjects. The darker shaded slices represent the proportion of correct responses. There appears to be a bias towards selecting haptic precedence with fast damping (first and second rows, second column) and audio precedence with slow damping (third and fourth rows, first column).

	Number Of Correct Responses	Number of Audio Precedence Selected
mean	21.75	23.58
max	26	31
min	17	16
std	2.86	4.52

Table 4.2: Number of correct responses, and number of audio precedence selected out of 48, compiled for all 12 subjects. Despite the lone subject with 17 correct responses, we cannot reject the hypothesis that the subjects performed at chance. The mean number of audio precedence selected (not necessarily correctly) suggests that the subjects did not have a bias towards choosing either an audio or haptic precedence.

and second rows, second column) and audio precedence with slow damping (third and fourth rows, first column). Varying the frequency does not appear to have as strong of an influence as varying the damping rate. We will return to this in the discussion section.

Table 4.2 shows the mean number of correct responses, along with the maximum, minimum, and standard deviation of correct responses out of 48. Table 4.2 also shows the number of audio responses selected. Four different subjects were responsible for each of the maxima and minima.

We tested the null hypothesis that the subjects perform at the chance level (each response is a pure guess) for each of the 12 subjects. By hypothesis, the mean number of correct responses $\mu = 24$ and the standard deviation $\sigma = 3.45$. Using the normal approximation to the binomial distribution we conclude that we can reject the hypothesis with a two-tailed test at the significance level $\alpha = 0.05$ only if the sample mean is outside the interval $\mu \pm 1.96\sigma = [17.21, 30.78]$. Except for the lone subject with 17 correct responses (Figure 4.9 third row, second column), we cannot reject the hypothesis that the subjects performed at chance. We note that a

one-tailed test may be more appropriate since we want to know if the subjects can detect the precedence better than chance. With a one tailed test we can not reject the hypothesis for any of the subjects.

The mean number of audio precedence selected in Table 4.2 suggests that the subjects did not have a particular bias towards choosing either an audio or haptic precedence. As mentioned earlier, subjects were not told that there were equal numbers of stimuli types nor were they told the number of repetitions in the experiment.

4.3.6 Discussion

The results indicate that 2ms is a valid lower bound for the perceptual tolerance of asynchrony for a contact interface like the AHI. Previous work by Rasch on asynchronies in performed ensemble music found deviations to lie between 30ms and 50ms. Although our study differed from his in the task, the device, and in the stimuli, our expectation for the result of our own study was that we would not have to reject the hypothesis. Nevertheless, we wished to verify that 2ms is a valid lower bound for a contact interface such as the AHI.

Figure 4.10 suggests that the rate of decay of the auditory stimulus could influence the user responses, independent of the actual stimulus precedence. Table 4.3 contains the six most extreme number of correct responses, listed across stimuli. Sounds that decay more slowly are perceived as preceding the haptic stimulus and sounds that decay quickly are perceived as lagging the haptic stimulus, independent of the actual order of stimulus presentation. An increase in the audio decay rate (sounds decaying more quickly) reduces the total energy and total duration of the audio signal. It is possible that the subjects are using decay rate or

Stimulus #	# Of Correct Responses	Stimulus Type
4	1	Audio+High+Fast
6	2	Haptic+High+Slow
11	10	Haptic+High+Fast
30	10	Haptic+Low+Fast
33	1	Haptic+Low+Slow
45	2	Haptic+Low+Slow

Table 4.3: The six most extreme values for number of correct responses, listed by stimuli number. The number of correct responses is out of 12. Stimulus type is listed in order of precedence, frequency, and decay.

total duration or total energy as a criterion for their response. The subject could be choosing the "loudest" signal (in terms of duration or energy) as the precedent stimulus. Another user study would be required to fully describe (and eventually understand) this effect.

The most glaring confound in our experiment design is that we used the same stimulus presentation order for all subjects. It is possible that hysteretic order effects may have unduly influenced the responses and in the worst case are so severe that we cannot generalize our results past the particular simulus order that we presented. If it becomes necessary to revisit this user study it will be necessary to retest a few subjects with a completely random stimulus order.

The other confound that might exist would be the lack of a blindfold on the participants. In the worst case we have a discrimination task between auditory, haptic, and visual events instead of just the two modalities that we wished to investigate. No explicit visual stimuli were provided to the subjects. Our personal experience in observing the subjects as they worked their way through the experiment was that many of them made a point of looking away from the AHI or closing

their eyes as they were striking the virtual wall. This isn't entirely surprising since it is well known that visual stimuli can easily dominate other perceptual modes. Looking away from the AHI could be a part of an information maximization strategy that attempts to save the most attention possible for discriminating between the auditory and haptic stimuli. In future experiments, we will likely blindfold the subjects to remove this confound.

4.3.7 Summary

Our user study served a few purposes. First, it was a preliminary effort to help verify that our system latency rendered perceptually simultaneous auditory and haptic events. Second, it helped us clear some of the technical brush before considering a more sophisticated user study to find the upper limit of synchronization tolerance. Finally, we were able to observe naive users approach and use the AHI. Their reactions were uniformly positive towards using the device. They worked with the rendered multimodal contact events as naturally as they would if tapping on a real surface. The AHI hardware and software design did not disrupt or scare the participants and executed reliably with no failures.

4.4 IRIS Demonstration

In May 2000 we took the AHI to the Institute for Robotics and Intelligent Systems (IRIS) conference in Montreal. The AHI was presented as a live hands-on demonstration for three consecutive days. At least 40 people tried the AHI. The general response was encouraging. Many tried the device out of curiosity, but several researchers familiar with either haptics or audio visited the demonstration and offered constructive criticism. This section describes the program modifications we made to

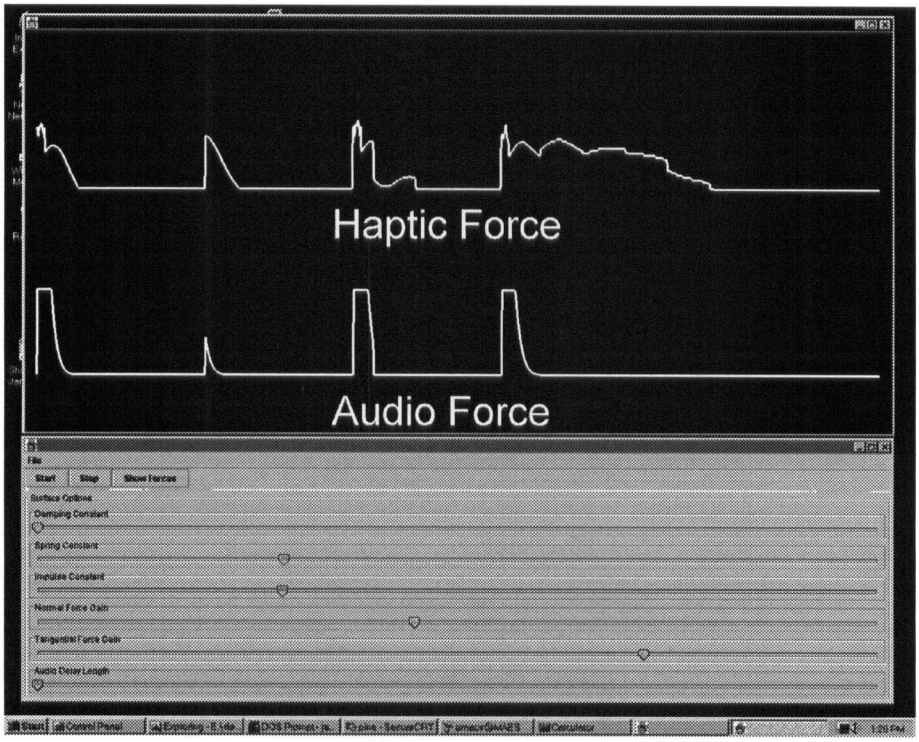

Figure 4.11: A screen capture from our Java GUI for visualizing contact forces and for interactive control of system parameters. The top window contains force profiles captured and rendered in real-time and the bottom window contains some buttons and sliders for starting the program, changing surface constants, etc. The audio force profile is a clipped and decayed version of the haptic force profile.

the AHI for this live demonstration and recounts the relevant comments we received.

One aim for the live demonstration was to implement a graphical user interface (GUI) to visualize the contact forces and to interactively control the system parameters. We implemented a Java based GUI that relied on native NT dynamically linked library (DLL) function calls to read and write data via shared memory to the MC8. Precision Microdynamics provides a set of simple memory read/write access functions that were wrapped inside the DLL for convenience. With this configuration we could adjust system parameters (surface constants, filtering options), load ACME sound models, and display haptic and audio force profiles in almost real-time. Figure 4.11 shows haptic and audio force profiles being captured. In this example, the audio force is a clipped and decayed version of the haptic force.

To demonstrate continuous coupled haptic and audio stimuli with the AHI, we also included the three friction models described in Section 3.4.2. Our observation during the IRIS demonstration was that viscous friction was haptically "believable", but did not produce ecologically valid sounds. This is likely due to our perceptual expectation that a sticky surface shouldn't make much noise. Convolving viscous friction forces with the audio impulse model sounds "rough", which is fair because our velocity measurements are first order differences. Differencing amplifies noise. Our experience with Salcudean's model is that it both felt and sounded noisy – again, we expect that this is a result of the velocity (a first order difference) amplifying system noise. One interesting observation is that people who tried this model commented on the noise in the audio signal rather than the noise in the haptic signal. Our best results were with Hayward's stick-slip model. In general Hayward's model felt and sounded much less noisy than either of the two previous friction models, likely because it only uses displacements (and not velocities) to compute forces. It

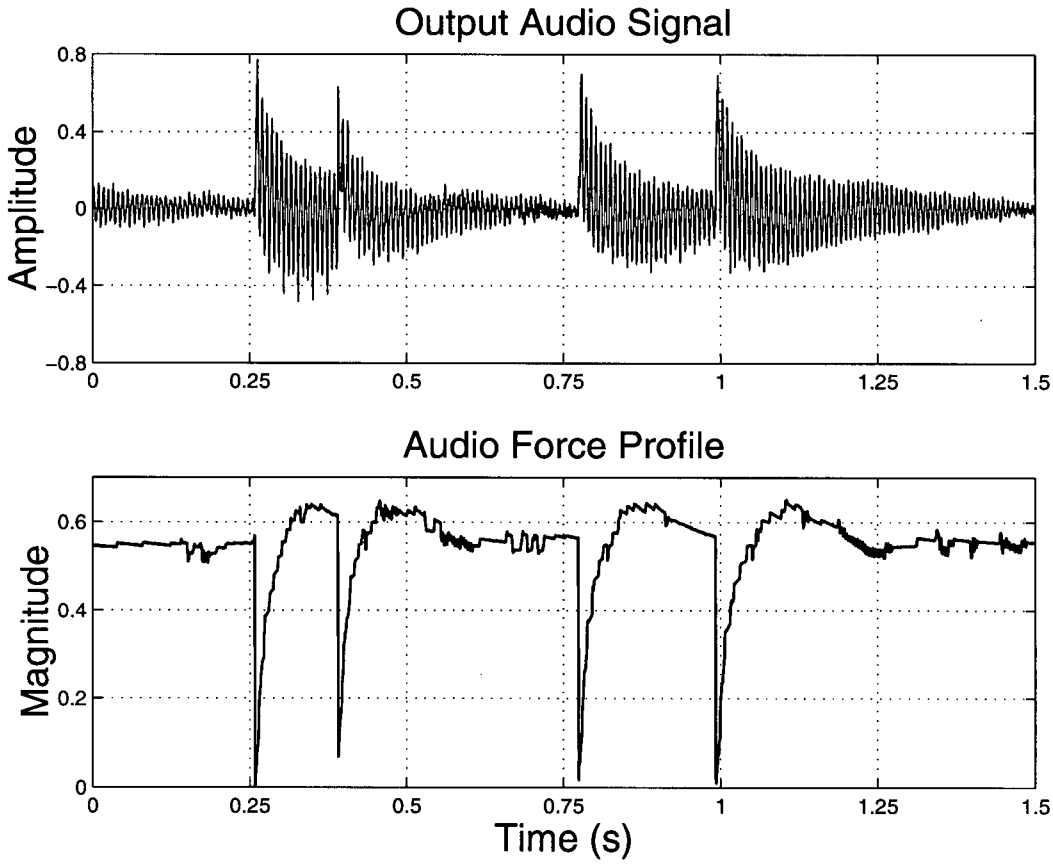

Figure 4.12: Output audio signal and audio force magnitude for interacting with the AHI and Hayward's stick-slip friction model. Audio force decay was enabled but only applied to the normal force component. Force interpolation was disabled. The force increases as the displacement between the real and proxy contact point increases and then discontinuously drops to zero during the slip phase.

generated the most interest from the IRIS participants. The model also exhibited good speed effects – scraping back and forth more quickly resulted in "squeakier" sounds.

Figure 4.12 shows captured audio signal and audio force magnitudes when interacting with the AHI and Hayward's stick-slip friction model. Audio force decay was enabled for this captured signal, but only applied to the normal force component. A force hysteresis is clearly visible. The force increases as the displacement between the real and proxy contact point increases (sticking phase) and then discontinuously drops to zero during the slip phase. These discontinuities in the audio force create impulses in the audio signal. We used a similar model of a brass vase in this example to the one in Figure 4.7.

There were some very valuable suggestions for improving the AHI's audio response. One suggestion was to make the audio force a logarithmically scaled version of the haptic force. This would compress the dynamic range of the auditory interaction. Linear scaling clips too quickly, resulting in no audible change in output volume even though the user is varying their strike force. Compressing the dynamic range allows for more headroom with high force interactions. Another suggestion was to include position information and graphics. Several people commented that their perception of the auditory stimulus would have improved if they could look at what they were striking and see the contact point moving over the surface of a virtual object. Both of these suggestions emphasize the importance of cross-modal similarity as discussed in Section 2.6. Improving the similarity between the dynamic range of auditory and haptic signals, and between the auditory and visual representation, will improve the usefulness and effectiveness of the AHI.

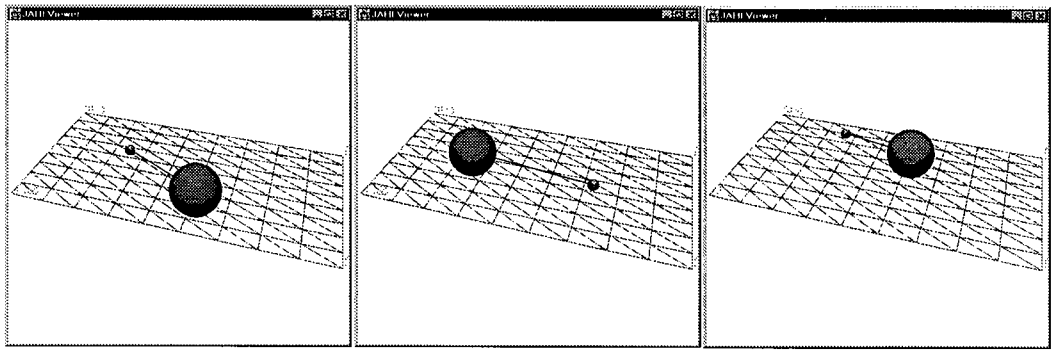

Figure 4.13: Screen captures from the rigid body dynamic simulation. The AHI pulls the ball across an inclined plane. The smaller sphere represents the AHI handle which connects to the center of the wooden ball by a spring. Moving the AHI handle in the plane applies a spring force to the wooden ball that is also felt as force-feedback to the user's hand.

4.5 Integration with a Rigid Body Dynamic Simulation

We have taken the first steps towards integrating the AHI with a rigid body dynamic simulation developed at UBC [Kry and Pai, 2000]. Our implementation task was to piggyback the AHI hardware and control code onto an existing Java implementation of the dynamic simulation. This section describes the high-level details of the simulation, the interface between it and the AHI, and presents the resulting audio signals synthesized by the algorithms in Section 3.3. Real recorded examples of spinning and scraping a "toonie" coin are included to help make a visual comparison.

Kry's simulation evolves a single contact between two smooth surfaces by treating the contact point as a generalized 5 DOF joint. The simulation maintains spatial velocities and wrenches at the contact point, then uses this information to time step the state by an explicit integration scheme. Running in Java on Windows NT the simulation can sustain refresh rates of about 20Hz.

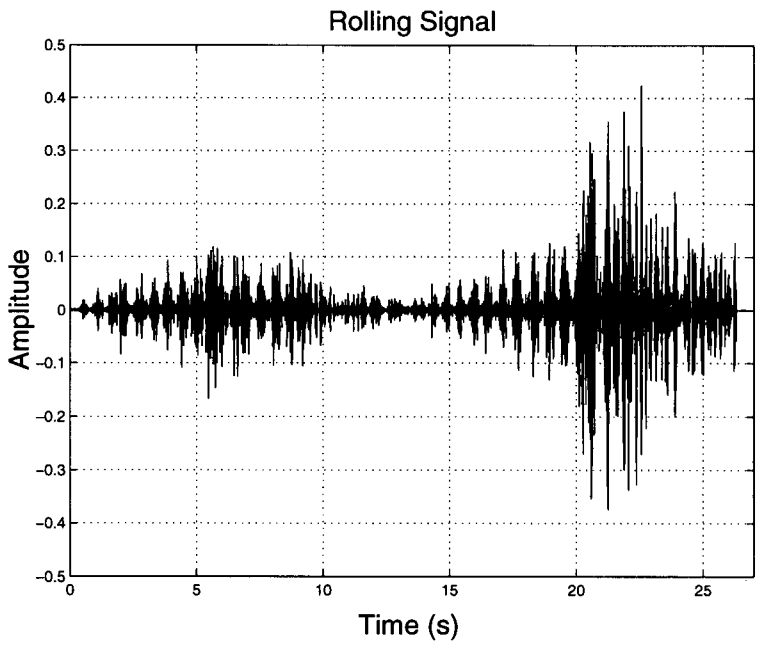

Figure 4.14: Synthesized signal of a rolling ball. The AHI drags the ball up an inclined plane. The ball is released at $t = 0$, rolls up the plane, stops at $t = 13$, then starts to roll down the plane. The AHI "catches" the ball around $t = 25$.

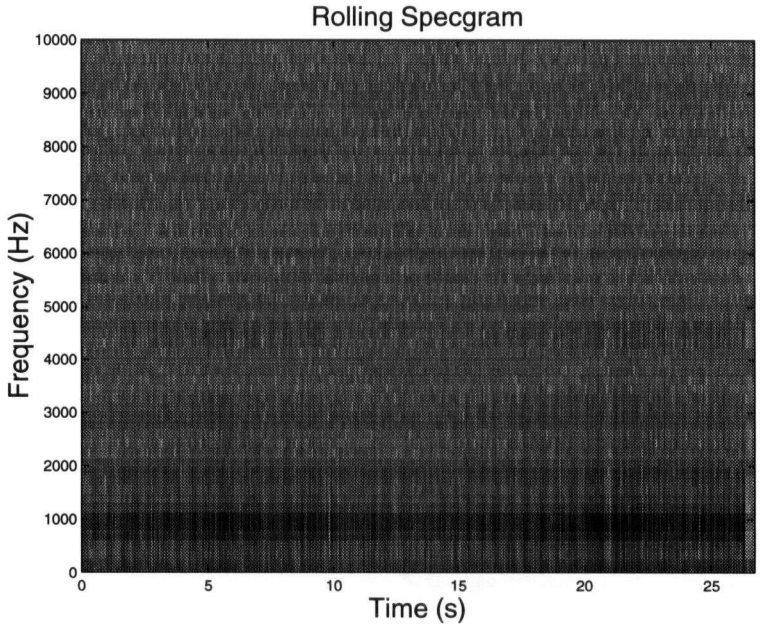

Figure 4.15: Spectrogram of the synthesized rolling signal in Figure 4.14. There is very little energy beyond the last mode at 4670Hz due to the Gammatone filtering.

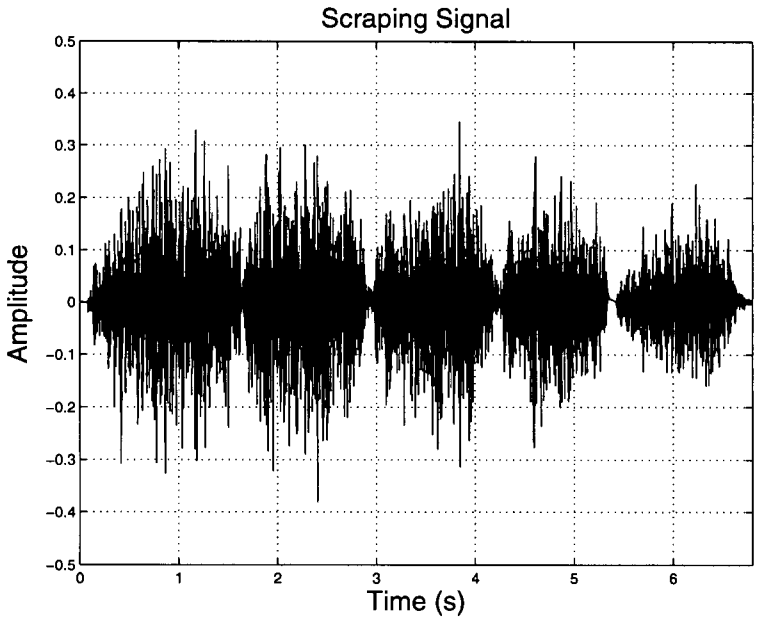

Figure 4.16: Synthesized signal of scraping a ball. The AHI pulls the ball back and forth across the plane.

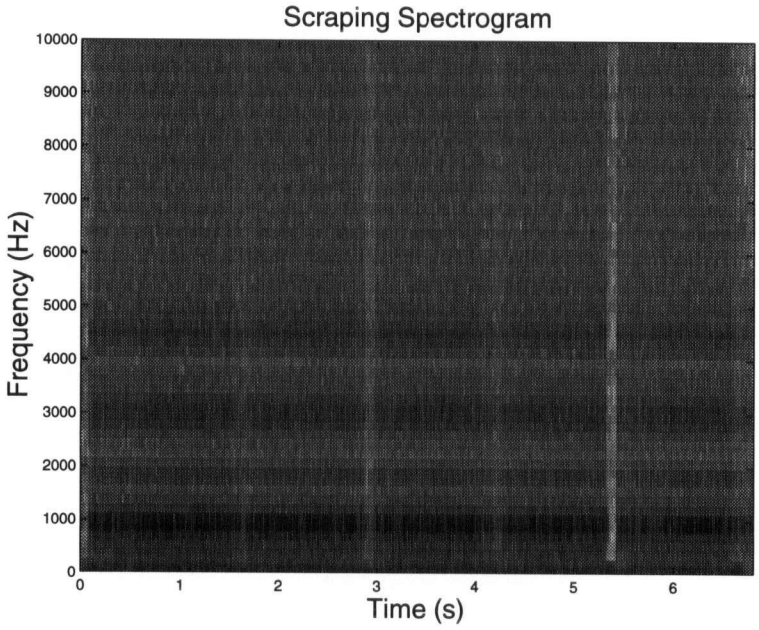

Figure 4.17: Spectrogram of the synthesized rolling signal in Figure 4.16. There is a lot of energy beyond the last mode at 4670Hz. The impulse response filter passes more high frequency energy than the Gammatone filter used for the rolling sound in Figure 4.14.

The simulation evolves single contacts between two smooth, convex, surfaces. For our experiments we limited the environment to an approximately spherical object in contact with a flat inclined plane. Figure 4.5 shows three screen captures of a "wooden" (textured) ball being pulled by the AHI across the inclined plane. The smaller sphere in the figure represents the AHI handle which connects to the center of the wooden ball by a spring. Moving the AHI handle in the plane applies a spring force to the wooden ball that is also felt as force-feedback to the user's hand. The Java GUI used for the IRIS demonstration was adapted to allow for reading and writing of parameters using shared memory and to provide GUI components for starting and stopping the simulation. Dynamic simulation parameters are read by the AHI control code at a 20Hz rate, nominally equal to the update rate of the Java simulation code.

By typing a key during the simulation the AHI can grab or release the ball. Figures 4.14 and 4.15 plot a synthesized audio signal and corresponding spectrogram of a rolling ball. In this example, the AHI is used to drag the ball up the inclined plane. Once the ball is released, it rolls up, stops, then rolls back down the inclined plane and is finally caught by the AHI and brought to a halt. The time signal starts when the AHI first releases the ball. We use an analytical audio impulse model like the one in Section 4.3 for an ideal struck bar except here we convolve forces with a Gammatone filter. The fundamental frequency is 350Hz. Typically, the angular velocity magnitudes vary between 0 and 15. In this particular example the parameters in Equation 3.8 were $p_{mean} = 5$ and $p_{sens} = 0$. Rolling impulses are computed during the haptic control loop so this value for the mean interarrival time translates into impulse arrivals at approximately 20Hz. The amplitude modulation depth and frequency scaling in Equation 3.9 are $a_{depth} = 0.71$ and $f_{roll} = 0.003$. The

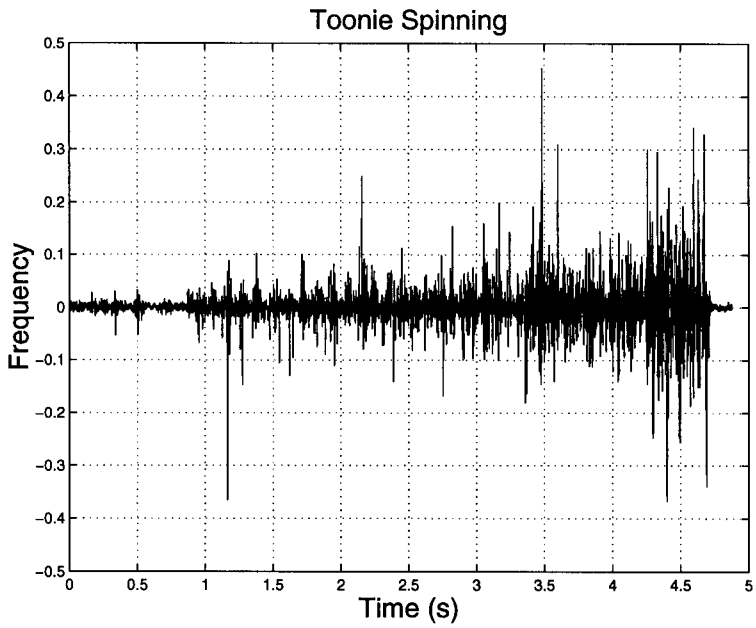

Figure 4.18: Recorded signal of a real toonie spinning on a desk. As the angular velocity of the toonie increases, the number of impacts increase as well. The toonie comes to rest at $t = 4.75$.

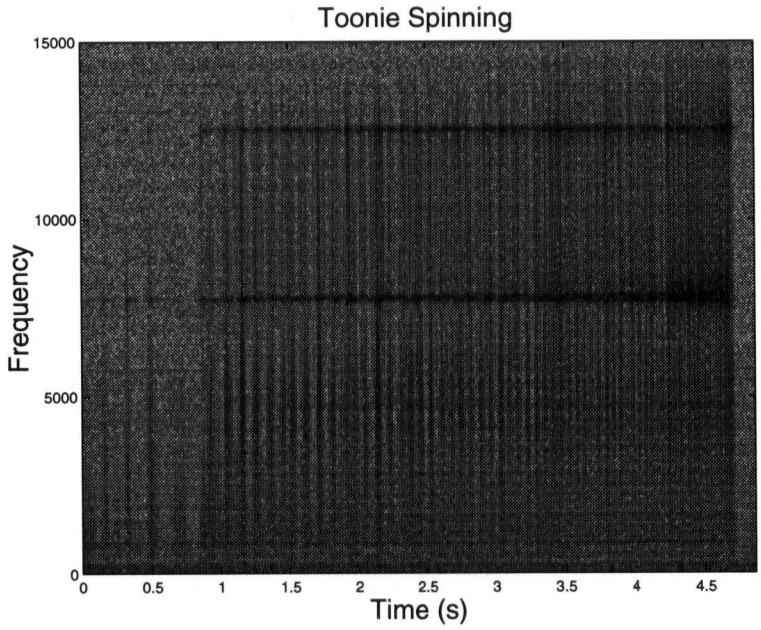

Figure 4.19: Spectrogram of the spinning toonie signal in Figure 4.18. The toonie has two resonant modal peaks at approximately 12.5kHz and 7.8kHz.

resulting signal sounds somewhat like a large ball rolling on a very rough surface.

It is also possible to use the AHI to scrape the ball back and forth across the plane. Figures 4.16 and 4.17 show a typical scraping interaction. The audio impulse model is the same as for the rolling example above, but without Gammatone filtering. Typically, the linear velocity magnitudes vary between 0 and 20. In this particular example the parameters in Equation 3.8 were $p_{mean} = 5.0$ and $p_{sens} = 0$. Scraping impulses are generated during the audio control loop, so the lowest value for the Poisson mean translates into impulse arrivals at approximately 4000Hz. The resulting signal sounds like very noisy scraping. The frequency content of the Poisson stream dominates the impulse response of the ball.

One issue for future consideration will be how to match rolling and scraping parameters to maximize their similarity. During a typical interaction the rolling and scraping signals are perceptually separate streams, almost as if they were coming from two different objects. It is not clear how to resolve this issue, but an obvious way to start is by examining some recordings of real signals.

Figures 4.18 through 4.21 show signals and spectrograms of rolling and scraping a real toonie on a smoothly textured computer desk. No attempt was made to isolate the signal from noise other than by maximizing the microphone proximity. We selected a toonie because it is a common object and it has two strong modal peaks at approximately 12.5kHz and 7.8kHz. These peaks are clearly visible in Figure 4.19. This figure also shows a decreasing mean for a stream of Poisson pulses as angular velocity increases (the toonie spins faster as it comes to rest). Individual Poisson pulses are not visible in the spectrogram of synthesized rolling in Figure 4.15. This is possibly due to different damping rates between the synthesized signal and the toonie.

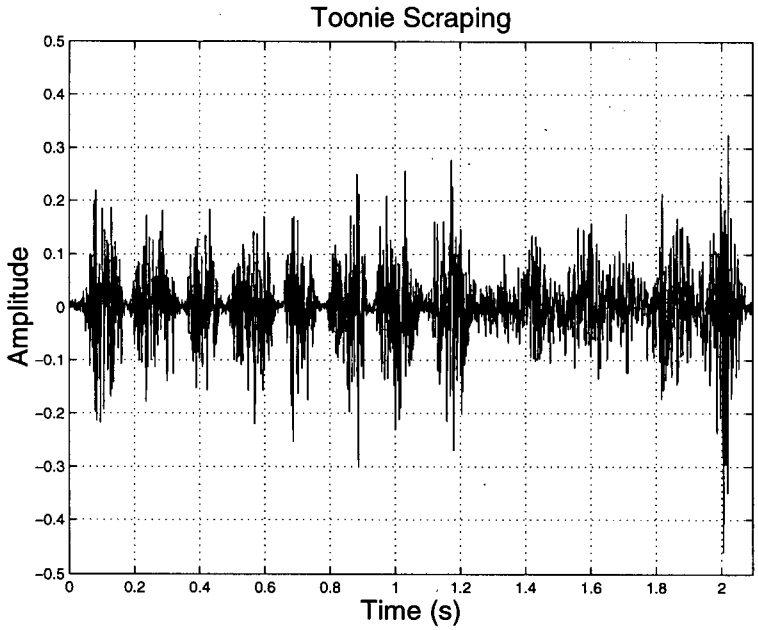

Figure 4.20: Recorded signal of a real toonie scraping on a desk.

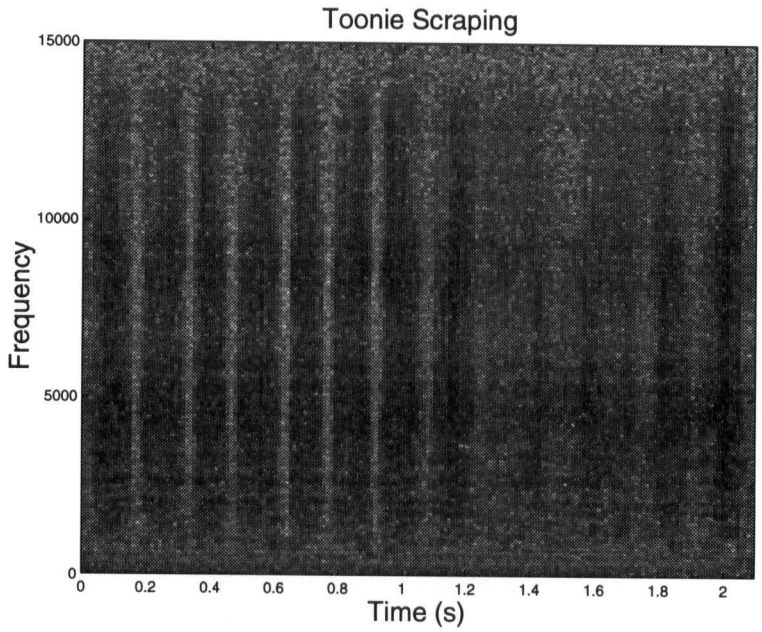

Figure 4.21: Spectrogram of the scraped toonie signal in Figure 4.20. Scraping blurs out the two modal peaks. The large amount of high frequency energy present corresponds reasonable well with the synthesized scraping in Figure 4.17.

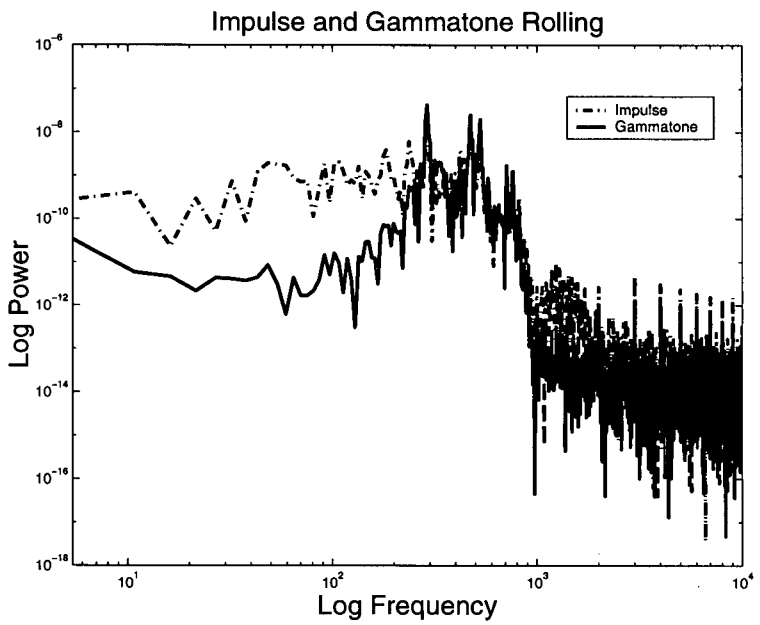

Figure 4.22: Power versus frequency for impulse and Gammatone response to a synthesized rolling force profile. The Gammatone filter (solid line) resonates more strongly at the peak frequency than the corresponding impulse filter (dashed line).

Scraping the toonie across the table blurs out the two modal peaks. Figure 4.21 shows energy evenly distributed up to just beyond the peak modal frequency of 12.5kHz. This energy distribution corresponds reasonably well with the spectrogram of the synthesized scrape in Figure 4.17.

Given that the modal peaks are more pronounced when a toonie spins than when it is scraped, and also that spinning is similar to rolling, the Gammatone filter could be worth keeping. Recalling Figures 3.3 and 3.5, the Gammatone filter resonates more strongly at the peak frequency than the corresponding impulse filter. It acts as more of a bandpass filter. Figure 4.22 plots the frequency responses of a rolling force profile filtered through an impulse filter and a Gammatone filter which both resonate at 350Hz. These filters behave as expected, but the frequency response

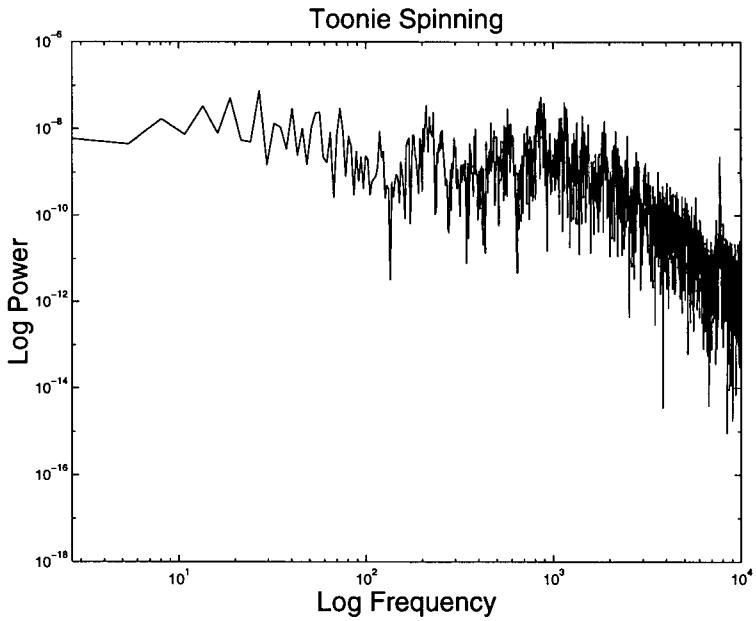

Figure 4.23: Power versus frequency for the recorded spinning toonie in Figure 4.18. This plot resembles the impulse filter response in Figure 4.22 more than the corresponding Gammatone filter response. The increased energy at low frequencies could be accounted for by including an impulse response model of the other surface (a desk) and the increased energy at higher frequencies by system noise due to the recording process.

of the toonie spinning in Figure 4.23 resembles the impulse filter response more than the Gammatone filter response. The mode at 7.8kHz rises out of the background noise. Perhaps the increased energy at low frequencies can be accounted for by including a model of the impulse response of the desk and the increased energy at high frequencies by ambient or internal noise.

4.6 Chapter Summary

This Chapter has presented three separate evaluations of the AHI. Based on the design criteria defined in Chapter 2 and the implementation described in Chapter 3 we have produced examples of dynamically synchronized auditory and haptic contact interactions, shown in Section 4.2. The AHI proved to be a reliable and useful tool for conducting our user study. Demonstrating the AHI to a critical audience helped reinforce our belief that the design choices outlined in Chapter 2 are worth keeping and significantly improving. Our preliminary work on integrating the AHI with a rigid body dynamic simulation produces a first approximation to rolling and scraping sounds based on user interaction. Ideas for improving the AHI and directions for future work will be considered in the next Chapter.

Chapter 5

Conclusion

5.1 Overview

This chapter will summarize the goals and achievements of this thesis, and will also outline a few directions for future work.

5.2 Goals

Our goal in developing the AHI was to implement a multimodal interface that could be precise enough for use in perceptual studies and interesting enough to motivate future research in simulating contact interactions. We believe that multimodal interfaces should allow a user to interact with virtual objects in the same way they do with everyday objects. Real-world interactions such as tapping and scraping produce synchronized and similar auditory and haptic stimuli. An ideal multimodal interface should be able to produce these stimuli. The AHI is the only human-computer interface we know of for providing closely coupled auditory and haptic stimuli with consistently low latency.

Our challenge was to balance the tradeoff between high realism and low latency. Widely available operating systems do not guarantee consistent performance for rendering audio. Due to the uncertain perceptual consequences of latency and asynchrony for rendering coupled audio and haptics, the central design criterion for the AHI was to maintain a low system latency without compromising the quality of the simulated contact interactions. Based on this design criterion, we selected a custom-built haptic device and dedicated hardware. We chose an audio synthesis algorithm that renders audio in real-time based on user interaction, responds to continuous input data, can represent the auditory properties of everyday objects, and can be parameterized based on measurements of everyday objects.

5.3 Achievements

To our knowledge, we have produced the first examples of dynamically synchronized auditory and haptic contact interactions. We also found the AHI to be a reliable and useful tool for conducting our user study. For our specific study, the results indicated that 2ms is a valid lower bound for the perceptual tolerance of asynchrony for a contact interface like the AHI. Demonstrating the AHI to a critical audience helped reinforce our belief that our design choices are worth keeping and significantly improving. Our preliminary work on integrating the AHI with a rigid body dynamic simulation produced examples of rolling and scraping sounds based on user interaction. There is no published work that we know of describing a device with similar capabilities.

The novelty of our particular implementation of the AHI and the necessity of using a specialized haptic device and dedicated hardware will fade with time. What we hope will remain is an appreciation for the potential of multimodal contact inter-

actions to enrich commercial, research, and entertainment applications. We expect that the audio synthesis techniques of Chapter 3 for realizing these interactions will also remain. In the next section we will consider a few of the future improvements and uses for the AHI.

5.4 Future Work

The following sections will detail two broad areas for future work. We touch upon three items to consider for improving the AHI: implementing collision detection, incorporating position dependence, and increasing the sophistication of our system models. In the last section we also consider further opportunities for using the AHI in perceptual studies of integrated audio and haptics, specifically for exploring multimodal texture and friction. One of the basic questions about cross-modal synchronization has been recently studied, but many questions about similarity and synchronization between dynamic vibration stimuli remain.

5.4.1 AHI Improvements

We still have not achieved even the basic requirements for the multimodal interface presented in the Introduction (Figure 1.1). The AHI needs a visual component. The PHANToM has 3D graphics and haptics, but poor audio support as discussed in Section 2.7. This audio support will improve drastically when Sensable provides Windows 2000 drivers and these should arrive by the end of 2000 [Sensable, 2000] [personal correspondence]. After audio support, collision detection is the next issue for porting the AHI algorithms to the PHANToM. Fast and accurate collision detection will determine the quality of continuous contact interactions. Collision detection will also allow us to replace the simple spring proxy with contact interac-

tions when using the dynamic simulation of Section 4.5. We should be able to strike the virtual ball with the AHI and see/hear it roll away as a result.

Researchers at the University of North Carolina have a publicly available package called H-collide that quickly and accurately implements point probe against 3D object collision detection [Gregory et al., 1999]. Their algorithms still produce a 1D penalty force magnitude for haptic feedback, so the audio synthesis techniques described in Chapter 3 will easily suffice for 3D contact events. Adapting the PHANToM environment to include H-collide involves no new research contributions, but extending the code to include 3D against 3D object collision will be non-trivial and will be necessary for representing probe-based interaction.

It is not clear how to effectively use position information for 2D or 3D continuous contact interactions like scraping and sliding. The ACME facility can provide us with multiresolution surface models that associate audio impulse response parameters with vertices. As the contact point moves across a triangular patch should just one audio impulse model be used or is it possible to linearly interpolate between sound models? Using position information naively will likely create audio dropouts as model parameters are discontinuously changed or beating phenomena (mismatch in frequency) or clicking (mismatch in phase) as model parameters are continuously changed and interfere with one another. Ideally, we would devise a multiresolution surface representation for auditory properties that could provide continuous data at any level of detail.

Previous work on automatic audio morphing relies on sophisticated frequency domain processing [Slaney et al., 1996]. Their technique uses 256 point windows for spectrogram calculation and they report that the initial spectrogram computation dominates the cost. At a sampling rate of 44.1kHz, a 256 point window is ap-

proximately 6ms – this is the minimum latency to expect from their method. By leveraging local coherence between sound models it could be possible to avoid the spectrogram calculations. Work on resolving this problem could start immediately with the AHI and then be ported to the PHANToM.

The AHI only models point against plane interactions. There are two important models that should be incorporated into the AHI control code: a dynamic model of probe-based interaction and a user model. In relation to Figure 1.2, these two models enhance our representation of the AHI contact interactions and of the user respectively. One grasps the handle of the AHI as one would grasp a pen or probe [1]. Adding the dynamics of the virtual tool or probe that mediates contact with the surface would increase the usefulness of the AHI for simulating teleoperated environments, and for perceptual studies. The user would be connected to the probe by a spring/damper combination (representing skin tension) and the probe would transmit forces and torques to the user's hand. The AHI already has a third motor for generating torques through the handle and collision detection between a probe and plane is trivial. This planar enhancement would not require much work, but the 3D extension would require an order of magnitude more work for collision detection and also a new PHANToM that can render the resulting torques.

There is a second dynamical system at work here: the human user. The unfathomable complexity of the human machine defies simplistic compartmentalization; nevertheless, we can identify three general areas to consider for modeling. The first is the tactile sense of touch at the finger pads, the second is the cognitive process that reacts and plans based on sensory input, and the third is the motor system that responds to mental commands.

[1] The PHANToM handle is actually a black plastic pen with the ink cartridge removed.

Nahvi has investigated the display of friction in haptic environments based on human finger pad characteristics [Nahvi and Hollerbach, 1998]. Their goal was to "imitate the real world movement of the human finger pad in the virtual environment" by extracting stick-slip model parameters from measurements of finger pad force versus displacement. Their results will be worth considering in more detail if the AHI algorithms are ported to the PHANToM. The PHANToM pen interface can be replaced by a fingertip "thimble" for one finger explorations by tapping and sliding. The first test that comes to mind for synchronized audio and haptics would be to simulate pushing a fingertip across a drum skin or glass surface. Representing and rendering the human finger pad will haptically enhance this sort of simulation and by convolution will likely result in an improved audio signal as well. Gillespie incorporated an impedance model based on empirical studies of limb dynamics and the time scale of volitional control to implement stiff virtual walls without chatter [Gillespie and Cutkosky, 1996]. They relied on numerical simulations to validate their results – they did not report online results with a working haptic interface. Reducing haptic chatter is important for the AHI because it leads to much more obvious and distracting auditory chatter (cf. the IRIS response to sonifying Salcudean's stick-slip model in Section 4.4). Both of the models just described were designed and implemented with only haptics in mind. We believe that haptic innovations that result in novel force profiles will translate into higher quality audio as well. Incorporating this existing research into the AHI could significantly extend the relevance of these models for multimodal simulation or suggest further improvements that include modeling auditory responses.

Modeling the human cognitive process that reacts and plans based on sensory input moves beyond mechanical models and into cognitive psychology and

psychophysics. The next subsection will return to the perception of multimodal similarity and synchronization first mentioned in Section 2.6. Our focus now will be on the utility of the AHI as a tool for perceptual studies of multimodal surface roughness.

5.4.2 Perceptual Studies and the AHI

Some recent psychophysical research has determined the perceptual tolerance for synchronization between auditory and haptic contact events [Levitin et al., 2000]. Our original plan was to conduct a very similar study ourselves. Although the first work on contact events has already been done, the AHI is well-suited to help establish similar perceptual tolerances for continuous audio and haptic interactions such as scraping and sliding over textured surfaces.

Levitin's practical motivation for this study was the same one we outlined in Section 2.6: an upper limit for reliable perceptual synchronization gives system designers a well-defined performance target. The 66ms 75% performance threshold is higher than what we expected partially based our own experience with the AHI.

There may be an important difference in synchronization tolerances between active haptic devices with motors such as the AHI and the PHANToM and passive devices such as the baton used in the Levitin study. The Pantograph motors do not operate quietly. Motor torques excite structural vibrations which produce sounds. It is known that the time resolution for successive audio clicks is on the order of 2ms [Green, 1973, Pierce, 1999] and that this value is largely independent of frequency. An intra-modal judgement would use both audio signals (one from the speakers, one from the motors) over the cross-modal judgement that compares the audio signal to the haptic signal, as shown in Figure 5.1. If this intra-modal judgement dominates

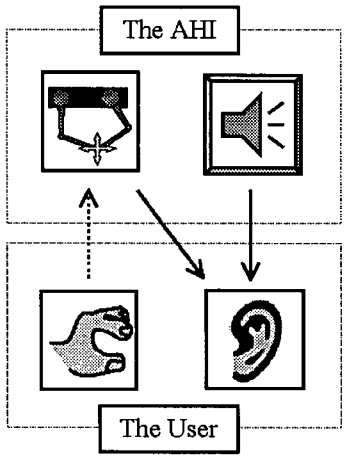

Figure 5.1: An intra-modal judgement using only the audio signals. Structural vibrations from the Pantograph device produce sounds that are heard by the user. If this sort of intra-modal judgement dominates the cross-modal judgement between haptics and audio it will likely be necessary to decrease the asynchrony between the modes.

the cross-modal judgement then it will likely be necessary to decrease the asynchrony to well below Levitin's reported figure. Identifying and controlling asynchronies for active devices like the AHI does not directly address cross-modal perception, but given the number of active haptic devices entering the market it would still be a valuable result to derive.

Our current simulation generates audio forces from haptic forces normal and tangential to a locally flat patch. Large scale surface features can consist of a collection of polygons which use our flat patch algorithms after collision detection. The spring/damper/impulse penalty method effectively and simply parameterizes the normal component as surface hardness. Limited versions of small scale tangential surface features – friction and texture – have been implemented and described in Sections 4.2 and 4.4. We want simple and effective parameterizations of the tan-

gential force component as friction and roughness variables that remain relevant for auditory perception.

Hayward's stick-slip model has has been applied to the real-time physical modeling of violin bow-string interactions [Serafin et al., 1999]. Bow-string interactions are some of the oldest studied examples of stick-slip friction. A good test for adding friction to our audio synthesis routine would be to simulate bowing a violin string and forcing it to resonate. Our AHI simulations of this sort of behaviour are not convincing yet – the audio signal in Figure 4.12 sounds more like a series of impacts than like the squealing we associate with stick-slip phenomena. We might also use his model for synthesizing rough textures by imposing Gaussian noise perturbation on either the separation between the proxy and real contact, or on the spring coefficient that produces friction forces. Our experience is that we will need another stage of force prefiltering to control the auditory roughness signal. Perhaps we could leverage work done in computer graphics on synthesizing fractional Brownian motion and fractals to gain more control over our signals [Ebert et al., 1998].

A more rigorous way to parameterize the tangential force component would involve active measurement of surface properties, coupling the ACME facility with recent work on friction identification for haptic display [Richard et al., 1999]. The same robotic arm that applies impulsive forces to our test objects could also be used to extract parameters from scraping and sliding motions. It should be possible, at least informally at first, to expand the existing work in friction identification to also consider the quality of the resulting auditory signal as a criterion for model selection and parameterization.

Previous studies on the influence of auditory stimuli on haptic perception have used audio samples or tones triggered by contact events. The study by Miner,

et al, used pitched tones and attack envelopes to simulate hard and soft sounds [Miner et al., 1996]. They found that "the auditory stimulus did not significantly influence the haptic perception". The study by DiFranco, et al, triggered audio samples of contact events they recorded by hand [DiFranco et al., 1997]. They found that "sound cues that are typically associated with tapping harder surfaces were generally perceived as stiffer". Both of these studies focus on the perception of hardness which is a function of force on the user's hand.

These studies do not explore the perception of surface roughness, which in addition to being a function of force, can also be a non-trivial function of interaction speed and surface geometry. When using a bare finger roughness perception is dominated by the spatial characteristics of the surface and by applied force [Taylor and Lederman, 1974]. Recalling Figure 2.2, roughness perception can be parameterized as lying primarily along the spatial/pressure axis, counter to our intuition that roughness perception should also depend on speed. In contrast, dynamic vibration effects as a function of speed are reported when interaction is mediated by a rigid probe [Lederman and Klatzky, 1999]. These effects are complex and not fully understood:

> [...] increasing speed tended to render surfaces as "smoother"; however, unlike earlier experiments that investigated the effect of speed using the bare finger, the current effect tended to reverse itself as the interelement spacing increased (pg. 17).

A texture model for any haptic interface (not just the AHI) could use these perceptual results to inform their implementation. However, because the AHI tightly couples the auditory stimulus with the underlying physical process of collision, a simple grid produces haptic and auditory textures that vary as a function of both

force and speed (Figure 4.7). Previous studies on perceiving auditory and haptic textures with the bare hand suggest that the subject will use the haptic texture before the auditory texture for discriminating roughness [Lederman, 1979]. The similar and synchronized stimuli rendered by the AHI could be used to study multimodal vibration effects when interaction is mediated by a rigid probe.

Devising and verifying these new multimodal friction and texture models will require several psychophysical studies. Regardless of the outcome, the AHI in its current state provides an excellent platform for experimenting with and understanding these models.

5.5 Conclusion

Designing compelling simulated environments is the high-level goal of this research. The representation and rendering of contact interactions comprises an essential component of any such simulation. Atomically representing a contact event as something that produces both sound and force helps integrate auditory and haptic stimuli. We believe this is a natural way to represent and simulate rigid contacts for interactive applications. This thesis has presented and evaluated an experimental multimodal interface for displaying realistic sounds and forces with low latency.

Bibliography

[Analog Devices, 1995] Analog Devices (1995). *ADSP-21000 Family C Runtime Library Manual.* Analog Devices, Norwood, MA, USA, third edition.

[Anderton et al., 1994] Anderton, C., Moses, B., and Bartlett, G. (1994). *Digital Projects for Musicians.* Amsco.

[Armstrong et al., 1994] Armstrong, B., Dupont, P., and de Wit, C. C. (1994). A Survey of Models, Analysis Tools and Compensation Methods for the Control of Machines with Friction. *Automatica*, 30(7):1083–1138.

[Bar-Joseph et al., 1999] Bar-Joseph, Z., Lischinski, D., and Werman, W. (1999). Granular Synthesis of Sound Textures using Statistical Learning. In *Proc. of the International Computer Music Conference*, Beijing, China.

[Brandt and Dannenberg, 1998] Brandt, E. and Dannenberg, R. (1998). Low-Latency Music Software Using Off-The-Shelf Operating Systems. In *Proc. of the International Computer Music Conference*, San Francisco CA.

[Cook, 1997] Cook, P. (1997). Physically Informed Sonic Modeling (PhISM): Synthesis of Percussive Sounds. *Computer Music Journal*, 21(3):38–49.

[Copley Controls, 1994] Copley Controls (1994). *Copley Model 306 PWM Brushed Servo Motor Amplifier.* Copley Controls, Westwood, MA, USA, first edition.

[Cota-Robles and Held, 1999] Cota-Robles, R. and Held, J. (1999). A Comparison of Windows Driver Model Latency Performance on Windows NT and Windows 98. In *USENIX Third Symposium on Operating Systems Design and Implementation.*

[Creative, 2000] Creative (2000). Soundblaster Live. http://www.soundblaster.com.

[DiFranco et al., 1997] DiFranco, D., Beauregard, G., and Srinivasan, M. (1997). The Effect of Auditory Cues on the Haptic Perception of Stiffness in Virtual Environments. In *Proc. of the ASME Dynamic Systems and Control Division.*

[Ebert et al., 1998] Ebert, D., Musgrave, F., Peachey, D., Perlin, K., and Worley, S. (1998). *Texturing and Modeling.* AP Professional, second edition.

[Fritz and Barner, 1996] Fritz, J. and Barner, K. (1996). Stochastic models for haptic texture. In *Proc. of the SPIE International Symposium on Intelligent Systems and Advanced Manufacturing*, Boston MA.

[Gaver, 1988] Gaver, W. (1988). *Everyday Listening and Auditory Icons.* PhD thesis, University of California, San Diego.

[Gaver, 1993] Gaver, W. (1993). Synthesizing Auditory Icons. In *Proc. of the ACM InterCHI*, pages 228–235, Amsterdam.

[Gillespie and Cutkosky, 1996] Gillespie, B. and Cutkosky, M. (1996). Stable User-Specific Rendering of the Virtual Wall. In *Proc. of the ASME International Mechanical Engineering Conference and Exposition*, Atlanta GA.

[Green, 1973] Green, D. (1973). Temporal Acuity as a function of frequency. *Journal of the Acoustical Society of America*, 54:373–379.

[Gregory et al., 1999] Gregory, A., Lin, M., Gottschalk, S., and Taylor, R. (1999). H-Collide: A Framework for Fast and Accurate Collision Detection for Haptic Interaction. In *Proc. of the IEEE Virtual Reality Conference*, Houston TX.

[Haptech, 2000] Haptech (2000). Haptech PenCAT/Pro. http://www.haptech.com.

[Hayward and Armstrong, 2000] Hayward, V. and Armstrong, B. (2000). *A new computational model of friction applied to haptic rendering*, volume 250, pages 403–412. Springer-Verlag.

[Hermes, 1998] Hermes, D. (1998). Synthesis of the sounds produced by rolling balls. Technical report, IPO Center for User-System Interaction, Eindhoven, The Netherlands.

[Hochberg, 1986] Hochberg, J. (1986). *Handbook of Perception and Human Performance*, chapter 24: Representation of Motion and Space in Video and Cinematic Displays. Wiley.

[Houben et al., 1999] Houben, M., Hermes, D., and Kohlrausch, A. (1999). Auditory perception of the size and velocity of rolling balls. Technical report, IPO Center for User-System Interaction, Eindhoven, The Netherlands.

[James and Pai, 1999] James, D. and Pai, D. (1999). ARTDEFO: Accurate Real Time Deformable Objects. In *Proc. of SIGGRAPH 99*, Los Angeles CA.

[Katz, 1989] Katz, D. (1989). *The World Of Touch.* Lawerence Erlbaum Associates.

[Klatzky et al., 2000] Klatzky, R., Pai, D., and Krotkov, E. (2000). Hearing Material: Perception of Material from Contact Sounds. *Presence (to appear).*

[Kry and Pai, 2000] Kry, P. and Pai, D. (2000). Fast Contact Evolution. Forthcoming.

[Kuper, 2000] Kuper, R. (2000). Audio I/O, Today and Tomorrow. http://www.cakewalk.com/DevXchange/audio_i.htm.

[Lederman, 1979] Lederman, S. (1979). Auditory texture perception. *Perception*, 8:93–103.

[Lederman and Klatzky, 1999] Lederman, S. and Klatzky, R. (1999). Perceiving Surface Roughness via a Rigid Probe: Effects of Exploration Speed and Mode of Touch. *The Electronic Journal of Haptics Research*, 1(1).

[Levitin et al., 2000] Levitin, D., MacLean, K., and Mathews, M. (2000). The Perception of Cross-Modal Simultaneity. *International Journal of Computing Anticipatory Systems (to appear).*

[Logitech, 2000] Logitech (2000). Logitech WingMan Force Feedback Mouse. http://www.logitech.com.

[Massie and Salisbury, 1994] Massie, T. and Salisbury, J. (1994). The PHANTOM Haptic Interface: A Device for Probing Virtual Objects. In *Proc. of the ASME Winter Annual Meeting, Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, Chicago IL.

[Maxon Motors, 1996] Maxon Motors (1996). *Maxon Model RE 035-071-33EAB 200A, Brushed DC.* Maxon Motors, Sachseln, Switzerland, first edition.

[McGurk and MacDonald, 1976] McGurk, K. and MacDonald, J. (1976). Hearing Lips and Seeing Voices. *Nature*, 264:746–748.

[Microsoft, 2000] Microsoft (2000). Microsoft SideWinder Force Feedback Joystick/Wheel. http://www.microsoft.com.

[Miner et al., 1996] Miner, N., Gillespie, B., and Caudell, T. (1996). Examining the Influence of Audio and Visual Stimuli on a Haptic Display. In *Proc. of the 1996 IMAGE Conference*, Phoenix AZ.

[Morse, 1968] Morse, P. (1968). *Theoretical Acoustics.* Princeton University Press.

[Murray et al., 1994] Murray, R., Li, Z., and Sastry, S. (1994). *A Mathematical Introduction to Robotic Manipulation.* CRC Press, Ann Arbor MI.

[Nahvi and Hollerbach, 1998] Nahvi, A. and Hollerbach, J. (1998). Display of friction in virtual environments based on human finger pad characteristics. In *Proc. of the ASME Dynamic Systems and Control Division*, Anaheim, CA.

[Pai et al., 2000] Pai, D., Ascher, U., and Kry, P. (2000). Forward dynamics algorithms for multibody chains and contacts. In *Proc. of the IEEE International Conference on Robotics and Automation*, San Francisco CA.

[Pai et al., 1999] Pai, D., Lang, J., Lloyd, J., and Woodham, R. (1999). ACME, A Telerobotic Active Measurement Facility. In *Proc. of the Sixth International Symposium on Experimental Robotics*, Sydney, Australia.

[Pierce, 1999] Pierce, J. (1999). *Music, Cognition, and Computerized Sound*, chapter 8: Hearing in time and space. The MIT Press.

[Precision MicroDynamics, 1999] Precision MicroDynamics (1999). *MCX-DSP-ISA Register Access Library and User's Manual.* Precision MicroDynamics Inc., Victoria, BC, Canada, fourth edition.

[Ramstein and Hayward, 1994] Ramstein, C. and Hayward, V. (1994). The Pantograph: a large workspace haptic device for a multi-modal Human-computer interaction. In *Proc. of the Conference on Human Factors in Computing Systems ACM/SIGCHI*, Boston MA.

[Rasch, 1979] Rasch, R. (1979). Synchronization in Performed Ensemble Music. *Acustica*, 43:121–131.

[Remington, 1987] Remington, P. (1987). Wheel/rail rolling noise, I: Theoretical analysis. *Journal of the Acoustical Society of America*, 81(6):1805–1823.

[Richard et al., 1999] Richard, C., Cutkosky, M., and MacLean, K. (1999). Friction Identification for Haptic Display. In *Proc. of the 1999 ASME IMECE*, Nashville TN.

[Richmond and Pai, 2000] Richmond, J. and Pai, D. (2000). Active Measurement of Contact Sounds. In *Proc. of the 2000 IEEE International Conference on Robotics and Automation*, San Francisco CA.

[Roads, 1996] Roads, C. (1996). *The Computer Music Tutorial.* The MIT Press.

[Salcudean and Vlaar, 1997] Salcudean, S. and Vlaar, T. (1997). On the Emulation of Stiff Walls and Static Friction with a Magnetically Levitated Input-Output Device. *Proc. of the ASME J. Dynamic Systems, Meas., Control*, 119:127–132.

[Senoner, 2000] Senoner, B. (2000). Linux Low-Latency Audio How-To. http://www.crosswinds.net/ linuxmusic/lowlatency.html.

[Sensable, 2000] Sensable (2000). PHANToM Haptic Device. http://www.sensable.com.

[Serafin et al., 1999] Serafin, S., Vergez, C., and Rodet, X. (1999). Friction and Application to Real-time Physical Modeling of a Violin. In *Proc. of the International Computer Music Conference*, Beijing.

[Siira and Pai, 1996] Siira, J. and Pai, D. (1996). Haptic Textures – A Stochastic Approach. In *Proc. of the IEEE International Conference on Robotics and Automation*, Minneapolis MN.

[Slaney, 1993] Slaney, M. (1993). An efficient implementation of the Patterson-Holdsworth Auditory Filter Bank. Technical Report 35, Apple Computer.

[Slaney et al., 1996] Slaney, M., Covell, M., and Lassiter, B. (1996). Automatic Audio Morphing. In *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Atlanta GA.

[Srinivasan and Basdogan, 1997] Srinivasan, M. and Basdogan, C. (1997). Haptics in Virtual Environments: Taxonomy, Research Status, and Challenges. *Comput. & Graphics*, 4(21):393–404.

[Steiglitz, 1996] Steiglitz, K. (1996). *A Digital Signal Processing Primer.* Addison-Wesley.

[Taylor and Lederman, 1974] Taylor, M. and Lederman, S. (1974). Tactile roughness of grooved surfaces: A model and the effect of friction. *Perception and Psychophysics*, 17(1).

[Thompson and Jones, 2000] Thompson, D. and Jones, C. (2000). A Review of the modelling of wheel/rail noise generation. *Journal of Sound and Vibration*, 231(3):519–536.

[Truax, 1988] Truax, B. (1988). Real-Time Granular Synthesis with a Digital Signal Processor. *Computer Music Journal*, 12(2):14–26.

[van den Doel and Pai, 1998] van den Doel, K. and Pai, D. (1998). The Sounds of Physical Shapes. *Presence*, 7(4):382–395.

[Wenzel et al., 2000] Wenzel, E., Miller, J., and Abel, J. (2000). A software-based system for interactive spatial sound synthesis. In *Proc. of the International Conference on Auditory Display*, Atlanta GA.

Appendix A

User Study

This appendix contains the master list of stimuli and all subject responses. It also includes the instruction sheet that was read to each participant in our user study.

A.1 Instructions

Before we start, I'm going to ask you a few questions.

- How old are you?
- Are you left or right handed?
- Do you have normal hearing?
- Do you have normal touch sensation in your hands?

In this experiment, you will strike a virtual object using this device. Striking the object will generate a force that you feel through the handle of the device, and sound that you hear through the headphones. You will strike this object many times. Each time either the sound you hear precedes the force you feel, or the force

precedes the sound. After each strike, you will tell me which stimulus you think came first: the force, or the sound. If you aren't sure which one came first, make your best guess.

The virtual object lies vertically, somewhere in the right half of the workspace. Before you strike the object, move the handle to the far left of the workspace. When I instruct you to strike the object, quickly move the handle left to right until you make a single contact, then return to the left side of the workspace. If you're right handed, use your right hand, etc.

I'll demonstrate a few times, then you can practice a bit before we start.

Hold the handle lightly, as you would hold a pen. Don't worry too much about your exact trajectory. Just move the handle quickly from right to left. Don't try to push through or break the object. Make a single contact and return.

Any final questions? You can stop the experiment anytime you like if you feel uncomfortable or would like to rest.

A.2 Stimuli and Responses

Master List of Stimuli
in order of presentation

1-	H H S	13-	H H F	25-	H L F	37-	H H S
2-	H H S	14-	A L S	26-	H L S	38-	H H F
3-	H L S	15-	A H S	27-	A L F	39-	H H F
4-	A H F	16-	A H S	28-	H H S	40-	H L F
5-	A L F	17-	A L S	29-	A H S	41-	A L S
6-	H H S	18-	A L F	30-	H L F	42-	H L F
7-	H H S	19-	H L F	31-	A H F	43-	A L S
8-	H L F	20-	A H F	32-	A L F	44-	H L S
9-	A L S	21-	A H S	33-	H L S	45-	H L S
10-	A H F	22-	A L F	34-	H L S	46-	A H F
11-	H H F	23-	H H F	35-	A H F	47-	A L F
12-	H H F	24-	A L S	36-	A H S	48-	A H S

Table A.1: The master list of stimuli in order of presentation. There are eight different types: Audio or Haptic precedence ("A" or "H"), High or Low frequency ("H" or "L"), and Fast or Slow damping ("F" or "S"). We used a total of 5 damped sinusoids with fundamental frequencies ω_1 of 1000Hz (High) and 316Hz (Low) and four higher partials. The damping coefficient τ_d was either 300 (Fast) or 3 (Slow).

Master List of Responses
for subjects 1 through 6 in presentation order

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
	a	a	**H**	h	h	a	a	a	h	h	**H**	**H**	a	**A**	h	h
S#1	**A**	h	**H**	h	h	h	**H**	**A**	**H**	**H**	h	a	**A**	**H**	h	h
	H	a	h	h	**H**	**H**	a	**H**	h	**H**	h	a	**H**	h	**A**	**A**
	H	a	a	h	**A**	**H**	**H**	a	**A**	h	**H**	a	a	**A**	h	h
S#2	h	h	a	h	**A**	h	a	**A**	a	a	**A**	a	h	**H**	h	h
	a	a	h	**A**	a	**H**	**H**	**H**	**A**	**H**	**A**	a	a	**A**	h	h
	H	**H**	a	h	**A**	a	a	a	**A**	h	**H**	**H**	**H**	h	h	**A**
S#3	h	h	a	**A**	**A**	h	**H**	h	a	**H**	**A**	a	**A**	**H**	h	h
	a	**H**	**A**	**A**	**H**	**H**	**H**	**H**	h	**H**	**A**	a	a	**A**	h	**A**
	H	a	a	h	h	a	**H**	**H**	h	**A**	**H**	a	a	h	**A**	h
S#4	h	**A**	a	**A**	**A**	h	**H**	**A**	a	a	**A**	a	h	**H**	**A**	**A**
	a	**H**	**A**	h	a	a	**H**	**H**	**A**	a	h	**H**	a	**A**	h	h
	a	**H**	**H**	h	**A**	a	a	a	**A**	h	a	a	**H**	**A**	**A**	**A**
S#5	**A**	**A**	**H**	h	**A**	h	a	**A**	**H**	a	h	**H**	**A**	**H**	h	h
	a	a	**A**	**A**	a	a	a	a	h	a	**A**	a	a	h	h	**A**
	a	**H**	**H**	**A**	h	a	**H**	**H**	h	h	**H**	a	**H**	**A**	**A**	**A**
S#6	**A**	h	**H**	h	**A**	**A**	**H**	h	**H**	**H**	h	a	**A**	**H**	h	h
	a	a	**A**	h	**H**	**H**	**H**	**H**	**A**	a	h	a	a	h	h	**A**

Table A.2: The master list of responses for subjects 1 through 6 in the order they were presented. Lower case "a" or "h" means an incorrect audio or haptic response and upper case bold face "A" or "H" means a correct audio or haptic response.

Master List of Responses
for subjects 7 through 12 in presentation order

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
S#7	a	**H**	a	h	h	a	a	a	h	**A**	**H**	**H**	**H**	**A**	h	**A**
	A	**A**	a	h	**A**	h	a	**A**	a	a	h	a	h	a	**A**	**A**
	a	**H**	h	h	a	a	a	**H**	h	**H**	h	**H**	a	**A**	h	**A**
S#8	**H**	**H**	a	h	h	a	a	a	h	**A**	**H**	**H**	**H**	h	h	**A**
	h	**A**	a	h	**A**	h	**H**	h	a	a	h	**H**	h	a	h	**A**
	a	**H**	h	h	a	a	**H**	a	h	**H**	h	a	**H**	**A**	h	h
S#9	a	**H**	**H**	h	**A**	a	**H**	**H**	**A**	h	a	**H**	a	h	**A**	**A**
	h	**A**	a	h	h	h	a	**A**	a	a	**A**	a	h	**H**	h	h
	a	a	h	**A**	a	a	**H**	**H**	**A**	**H**	h	**H**	a	h	**A**	**A**
S#10	**H**	**H**	**H**	h	h	a	a	**H**	**A**	**A**	**H**	**H**	**H**	**A**	h	**A**
	A	h	**H**	**A**	**A**	h	a	h	a	a	h	a	**A**	**H**	h	**A**
	a	a	**A**	**A**	a	**H**	**H**	a	**A**	a	h	a	a	**A**	h	**A**
S#11	a	a	a	h	h	a	a	**H**	**A**	**A**	**H**	a	a	h	**A**	h
	h	**A**	**H**	**A**	**A**	h	a	h	a	**H**	**A**	**H**	**A**	**H**	h	**A**
	a	**H**	h	**A**	a	a	**H**	a	**A**	**H**	**A**	**H**	a	h	**A**	h
S#12	**H**	**H**	**H**	h	**A**	**H**	**H**	a	**A**	h	**H**	a	**H**	h	h	**A**
	h	h	a	h	**A**	h	a	h	**H**	**H**	**A**	a	h	**H**	**A**	h
	a	**H**	h	**A**	a	**H**	a	**H**	h	**H**	h	a	a	h	**A**	h

Table A.3: The master list of responses for subjects 7 through 12 in the order they were presented. Lower case "a' or "h" means an incorrect audio or haptic response and upper case bold face "A" or "H"means a correct audio or haptic response.

Master list of Responses
for subjects 1 through 4 by factor

	AHF	ALF	AHS	ALS	HHF	HLF	HHS	HLS
S#1	h	**A**	**A**	**A**	**H**	**H**	**H**	**H**
	h	h	**A**	**A**	**H**	**H**	a	**H**
	h	h	h	**A**	**H**	**H**	a	**H**
	h	h	h	h	**H**	**H**	a	**H**
	h	h	h	h	**H**	**H**	a	a
	h	h	h	h	a	a	a	a
S#2	**A**	**A**	**A**	**A**	**H**	**H**	**H**	a
	h	**A**	**A**	**A**	**H**	**H**	**H**	a
	h	h	h	**A**	**H**	**H**	**H**	a
	h	h	h	**A**	a	a	a	a
	h	h	h	**A**	a	a	a	a
	h	h	h	h	a	a	a	a
S#3	**A**	**A**	**A**	**A**	**H**	**H**	**H**	**H**
	A	**A**	**A**	**A**	**H**	**H**	**H**	**H**
	A	h	**A**	h	**H**	**H**	**H**	a
	h	h	**A**	h	**H**	a	a	a
	h	h	**A**	h	**H**	a	a	a
	h	h	h	h	**H**	a	a	a
S#4	**A**	**A**	**A**	**A**	**H**	**H**	**H**	**H**
	A	**A**	**A**	**A**	**H**	**H**	**H**	**H**
	A	**A**	h	h	**H**	**H**	a	a
	A	h	h	h	a	a	a	a
	A	h	h	h	a	a	a	a
	h	h	h	h	a	a	a	a

Table A.4: The master list of responses for subjects 1 through 4 by factor. Lower case "a" or "h" means an incorrect audio or haptic response and upper case bold face "A" or "H" means a correct audio or haptic response.

Master list of Responses
for subjects 5 through 8 by factor

	AHF	ALF	AHS	ALS	HHF	HLF	HHS	HLS
S#5	**A**	**A**	**A**	**A**	**H**	**H**	**H**	**H**
	h	**A**	**A**	**A**	a	**H**	**H**	a
	h	h	**A**	**A**	a	**H**	a	a
	h	h	**A**	**A**	a	a	a	a
	h	h	**A**	**A**	a	a	a	a
	h	h	**A**	h	a	a	a	a
S#6	**A**	**A**	**A**	**A**	**H**	**H**	**H**	**H**
	A	h	**A**	**A**	**H**	**H**	**H**	**H**
	h	h	**A**	**A**	**H**	**H**	**H**	a
	h	h	**A**	h	**H**	**H**	a	a
	h	h	**A**	h	**H**	**H**	a	a
	h	h	h	h	a	a	a	a
S#7	**A**	**A**	**A**	**A**	**H**	**H**	**H**	**H**
	A	**A**	**A**	**A**	**H**	**H**	a	**H**
	A	h	**A**	**A**	**H**	a	a	a
	h	h	h	h	a	a	a	a
	h	h	h	h	a	a	a	a
	h	h	h	h	a	a	a	a
S#8	**A**	**A**	**A**	h	**H**	**H**	**H**	**H**
	A	**A**	**A**	h	**H**	a	**H**	**H**
	h	h	h	h	**H**	a	**H**	a
	h	h	h	h	**H**	a	a	a
	h	h	h	h	**H**	a	a	a
	h	h	h	h	a	a	a	a

Table A.5: The master list of responses for subjects 5 through 8 by factor. Lower case "a" or "h" means an incorrect audio or haptic response and upper case bold face "A" or "H" means a correct audio or haptic response.

Master list of Responses
for subjects 9 through 12 by factor

	AHF	ALF	AHS	ALS	HHF	HLF	HHS	HLS
S#9	h	**A**	**A**	**A**	**H**	**H**	**H**	**H**
	h	**A**	**A**	**A**	**H**	**H**	**H**	**H**
	h	**A**	**A**	**A**	a	**H**	a	a
	h	**A**	**A**	h	a	**H**	a	a
	h	h	h	h	a	a	a	a
	h	h	h	h	a	a	a	a
S#10	**A**	**A**	**A**	**A**	**H**	**H**	**H**	**H**
	A	h	**A**	**A**	**H**	**H**	**H**	a
	A	h	**A**	**A**	**H**	**H**	a	a
	A	h	**A**	**A**	**H**	a	a	a
	h	h	**A**	h	**H**	a	a	a
	h	h	h	h	a	a	a	a
S#11	**A**	**A**	**A**	**A**	**H**	**H**	**H**	**H**
	A	**A**	**A**	**A**	**H**	**H**	a	**H**
	h	**A**	**A**	**A**	a	**H**	a	**H**
	h	**A**	**A**	h	a	**H**	a	a
	h	h	h	h	a	a	a	a
	h	h	h	h	a	a	a	a
S#12	**A**	**A**	**A**	**A**	**H**	**H**	**H**	**H**
	h	**A**	**A**	h	**H**	**H**	**H**	**H**
	h	**A**	**A**	h	**H**	**H**	**H**	**H**
	h	h	h	h	a	**H**	**H**	a
	h	h	h	h	a	a	a	a
	h	h	h	h	a	a	a	a

Table A.6: The master list of responses for subjects 9 through 12 by factor. Lower case "a" or "h" means an incorrect audio or haptic response and upper case bold face "A" or "H" means a correct audio or haptic response.