Simulation of Stroke-able Percussion Surfaces Using A Haptically Extended Mouse

Mediaology MSc Thesis
Sound and Music Speciality

Ben Cahill
Aalborg University
Summer 2014

Aalborg University Copenhagen

Semester: 10

Title: Simulation of Stroke-able Percussion Surfaces Using A Haptically Extended Mouse

Project Period: January - May 2014

Aalborg University Copenhagen
A.C. Meyers Vænge
2450 København SV, Denmark

Semester Coordinator:

Secretary:

Supervisor(s): Stefania Serafin

Members: Ben Cahill

Copies: 3
Pages: 70
Finished: 28 May 2014

Abstract:

Tactile control systems for music interaction are abundant in the academic world. Many commercial devices exist that can simulate discrete boundary type interactions well and that do a passable job simulation other interactions.
This project examines the concept of adapting a familiar physical interface, the computer mouse, for use in a music interaction where continuous input is required.
A general design for haptic interaction is presented as a function of the users input forces which can be easily manipulated to create the illusion of different types of surface.
A control system using audio signals is implemented by compromising between several pertinent factors relating to performance, practicality and noise.
The system shows potential for a unique, yet familiar, interface for music expression.

Contents

Introduction

Motivation

The motivation for this work comes mainly from two places. Firstly, my love of all things musical; particularly performance, composition and improvisation, and secondly, my curiosity about how a person can be manipulated into experiencing something unexpected with sensory illusions. Electronics, interaction and software design are areas of interest in which I wish to broaden my skill-set and will use this work as means to do so.

Combined with my parents' record collection and TV, two early experiences made me want to make music.

I was about 8 years old and in hospital getting surgery for my sinuses. One of the nurses came to check on me and started drumming on the bed, really wailing. He told me he was a drummer and played in a band. It amazed me that a living, breathing, right-in-front-of-me person could play music, before that music was something on TV or on a record.

About a year or two later my brother and I got a tape recorder each for Christmas and consequently spent hours making tape-to-tape dubs with the built-in speakers and microphones.

My dad used to pester me to take up the guitar and listen to Led Zeppelin. I eventually did both, later progressing to four-track cassette, open reel, hard disk and finally computer recording many years later. It is *astounding* to me to see how far music making hardware has progressed in my lifetime, and to bear witness to the digital revolution. And it is a revolution; musicians can do things now that were unimaginable 20 years ago, new musical genres have been born and performance and composition have been forever changed.

One aspect of digital music making is, in my view, underrepresented in commercially available products and as a part of the digital music culture; haptics. There is a growing body of academic work which shows great progress in the area of musical haptics but little mainstream adoption of artificial haptics. There is also comparatively little non-academic DIY haptics, as opposed to the burgeoning modding and programming scenes. Why is that? Perhaps the instruments do not meet the player's demands; perhaps the effect is too unrealistic, perhaps haptics are too difficult to work with. There are as many questions as there are instruments. My intention with this work is to examine my own question about musical haptics and make a positive contribution to the overall body of work.

Background

When life first emerged from the primordial soup, it is likely it did so with a “splat” and initiated a percussive tradition amongst living beings. Today, we observe the chest-beating of apes as a display of strength and virility. In much the same way, man adopted this in battle by using drummers to intimidate the opposing forces, and many primitive cultures use percussion instruments in ceremonial events as an indicator and instigator of mood. It is perhaps likely that the first instrumental music made by man was percussive; the tools to make a variety of percussion instruments are found abundantly in nature and may be made with little tooling. This is difficult to state empirically as there is some ambiguity regarding fossils of percussion instruments; for example, a bone flute is obviously a bone flute by design, but is a woodblock and stick just a couple of pieces of fossilised wood that could have been something else?

As cultures advanced technologically, new tempered instruments were built that we able to produce discrete tones which offered musicians and composers[1] of the time a wider range of possibilities. As modern music advanced in the West, the prominence of percussive music declined and even folk traditions became synonymous with stringed and blown instruments, although some traditions use percussion in ensemble i.e. the Irish bohdran.

African folk tradition has a strong emphasis on percussion, and the cross-pollination of Africans with other cultures via the slave trade boom in the seventeenth to nineteenth century led to a transformation of Western music. A new type of music was born, influenced by both the existing folk tradition and by the imported rhythms; from this seed, we have gone from field songs to Jazz, Rap, Death Metal and Techno and its many, many derivatives. “Traditional” Western music still exists in the form of the styles from the Baroque period and earlier although much of this is now performed on modern instruments. Rhythmic, African-derived music is the order of the day for the majority of musicians nowadays.

To summarise; we musicians have been hitting things to make music for a long time and have done so under a huge range of circumstances and never have more of us done it than now. The need for battle musicians has fortunately subsided, and most music is now made for pleasure or recreation, with a small percentage of professionals doing it for the money. A huge industry has grown in tandem with the popularity of African-derived music and great technological advances have been made. With the era of amplified and recorded music, we have witnessed how technology changes the way people hear music, and this has in turn affected the way music is composed. Think of how the microphone informed the “crooner” vocal style, how the electric guitar kick-started Rock’n’Roll and, more recently, how sampling launched a myriad of genres.

We have always lived in a cycle between music style and the limit of current technology; as technology advances, music styles emerge that test the boundaries of the new technology; compare harpsichord music to early piano music, the piano allowed dynamic control and composers exploited this to create more expressive music. With the advent of low-cost digital technology, that cycle is shorter than ever

[1] The term ”composer” is loose here, the idea of ownership of a piece of music is relatively new. Typical folk traditions pass down songs from generation to generation with no notation, and the piece is reinterpreted by each subsequent generation.There is no individual sense of ownership, but perhaps a common one.

before; new instruments, both facsimiles of existing ones and entirely new ones, are being made all the time by amateurs and commercial companies and new forms of electronic music are constantly emerging.

Many modern composers and musicians choose to record in the digital domain due to the many conveniences it offers over typical analogue recording methods; i.e. higher fidelity, higher track counts, easier editing etc. Commercial instrument manufacturers have implemented the MIDI control protocol in a variety of instruments for 35 years but the keyboard remains the most widely used controlling mechanism. This is perhaps due to the general familiarity with the piano, and the vast array of student literature and music available for the instrument. The piano keyboard represents a compromise in many ways; notes cannot be sustained indefinitely and proper crescendo on one note is impossible, notes cannot be bent and glissando is not possible.

Systems that provide an alternative to the keyboard layout have failed to gain popular acceptance, i.e. the Monome-type matrix controllers or ribbon controllers , but percussion controllers have proved to be relatively popular. Electronic drum kits have improved dramatically in the last 10 years and are now used in place of acoustic drums by many drummers. The high end models offer multi-zone user-tensioned drum heads with rim and side-stick articulation and realistic feeling cymbals, and the sounds are usually high-quality multi-samples with some procedural filtering (i.e. Roland V-Drums XL series). The launch of the all-in-one MPC sampling workstation by Akai in 1988 popularised the 4*4 matrix of soft buttons that has become a stand for drum machines since. These controllers address the musician's need to control transient percussion where the physical action is to strike; drums, blocks, cymbals, bells, but the stroke-able subset of percussion instruments is neglected.

Figure 1 Music controllers. (Top) Ribbon controller, (left) Electronic Drums, (right) Akai MPC Drum pad and encoder unit.

As far as I am aware, there is no current controller dedicated to stroke-able percussion and any instrument plug-ins that simulate stroke-able percussion do so by mapping samples chromatically across the keyboard and have some tempo synchronisation features, i.e. the user triggers samples at certain bar divisions to compose a part, but may not play every note. The software then does some quantisation to lock the part to tempo. This solution works, but the visceral feeling of playing the instrument is lost and my intention with this work is to develop a stroke-able percussion controller and software system that allows a more natural interaction, analogous to the real-world experience.

The Haptic Sense

The haptic sense how we interact physically with the world around us, by touching, lifting, standing or sitting we build a picture of an object or environment based on how they feel. Touch can be pleasant like stroking an animal soft fabric, but it may also be unpleasant or even painful, like being burnt. We can ascertain a lot of information through our haptic sense; the weight of an object, the stiffness and roughness of a surface and the size of an object can all be estimated quite accurately with only haptic input.

Our haptic sense relies, like our other senses, on a combination of information gathered through sensing organs and an already built "image" of the world. We experience all of this through our skin, the body's largest organ, and our muscle system. Each of these sensing organs gathers a different type of haptic information.

We are not born with a fully developed awareness of our body. If you have ever seen an infant feeding itself, you will know that getting one spoon of food out of ten into their mouth is a good average. As odd as it may sound, the infant does not intuitively know how to raise its hand to its mouth, and this is a skill that must be developed over many years and with a great deal of practice. The development of the proprioceptive sense is what allows us to learn what seem like initially trivial skills (walking, eating, throwing) to the mastery of any physical activity (dancing, swimming, playing an instrument, typing).

The surface of the skin can resolve ***cutaneous*** haptic events. These are events in which the surface of the skin is displaced at a low level, and allows a person to feel small indentations in a surface if she is actively exploring one. It also provides information about being touched, discretely by another body, or continuously by for example clothing or when submerged in water. The distribution of these sensing organs on the body is not uniform; the lips and fingertips have a higher concentration than any other part of the body and are therefore the parts of our bodies most sensitive to cutaneous events.

Our muscle system resolves ***proprioceptive*** events. These are events that are associated with how we perceive our body in space, and as an extension of that how we perceive objects. As an example, think of the difference between standing and crouching, the cutaneous feedback is the same; both feet are being pushed by our own weight into the ground. We perceive our body as being in a different position without having to see it. Our legs are bent and there may be a shift in our centre of gravity and we instinctively know where our limbs are in relation to one another. Another proprioceptive event is experiencing the weight or size of an object, we feel the energy required to lift an object and evaluate its weight from that. Likewise with shape, we can tell how large an object is by how much space our arms tell us it fills. We can also tell how stiff an object is by the property of physical resistance, we can tell both how much force is required to displace something and how much it is displaced by.

Each of these types of sense receptors perform differently based on the stimuli they receive. The skin-based cutaneous receptors are able to determine how rough a surface is but only when the vibration frequency is over the threshold and amplitude for proprioceptive events. That means that the sensors for the resolution of high-frequency events are called into play when the vibrations are fast enough to determine a continuously rough surface, as opposed to a discretely distressed surface. [Hollins] shows how fatiguing the fingertip with a high frequency vibration (100Hz) diminishes the sensitivity to surface roughness but applying a low frequency vibration (10Hz) has no effect. Hollins describes the sensitivity as “severely hampered”, and presents an important point to developers of haptic interfaces; the user will adapt to continuous high frequency stimulation after a time, as the sensitivity of the skin is diminished. If, for example, the fingertip is rubbed on a rough surface for a length of time, the skin will become numb and desensitised to further stimulation. Much in the same way as exposure to continuous loud sound will cause the sense of hearing to become numb. It should therefore be taken into account when designing haptic devices that long exposure to high frequency vibration will limit the user’s ability to discriminate between small changes in cutaneous amplitude.

It is the reliance of the brain on prior knowledge that allows the sense to be cheated and haptic illusions created. An example could be crossing your fingers and touching the tip of your nose. The cutaneous feedback on the outside of the fingers does not make sense with the proprioceptive rubbing motion. The brain expects displacement on the inside of the fingers and the resulting sensation is that of having an “inside out” nose.

All haptic events in the real world are comprised of both cutaneous and proprioceptive components as a haptic event always involves some degree of friction. Friction is everywhere, all the time, and we experience it in many different forms, and for our entire lives. This means that our haptic sense, like our auditory sense is always engaged and cannot be voluntarily switched-off.

If we think of multisensory interactions, the possibility for incongruity is great. The haptic and auditory senses are closely related as they both parse mechanical vibrations to the brain and are always active. The visual and haptic senses are not as interdependent as the audio and visual senses, perhaps because we require physical contact with a stimulus to make haptic sense of it. When watching a film, we put the audio and video tracks together to create the illusion of people speaking on screen, but we do not expect a photograph or the surface of a screen to have a texture representative of what is being depicted.

In much the same way as discrete audio events merge into continuous tones when the separation between them in the temporal domain is very small, haptic events are perceived as continuous vibrations when the separation is below 30ms[Gescheider]. This is considerably lower than the equivalent ability of the auditory sense (1 -2ms). This again may be because audio and haptic events are very often perceived at close proximity in nature. One cannot experience a haptic event from a distance unless it is of great magnitude, whereas we can see and hear at a distance albeit with a delay in the sound. This underlines the importance for tight synchronisation in simulations of audio-haptic systems.

[Adelstein] shows how a subject’s tolerance for asynchrony between audio and haptic events is around 24ms when comparing two consecutive strike simulations. The experimenters also note one participant with a very high sensitivity to audio-haptic asynchrony which may not necessarily be unusual, as they may have had for example musical training which would have focussed their audio and haptic senses. Subjects were able to discriminate disparate stimuli consistently and most performed well.

Comparing the performance of the haptic sense to our other senses shows that it is superior in some ways but lesser in others. In his study of the legibility of Braille and tactile Roman letters, [Loomis] asks subjects to differentiate between characters of varying point-density to determine tolerances for spatial acuity. The findings of these experiments are that the subjects are only able to resolve differences of above 1mm using the fingertips. Loomis borrows a signal processing term, low-pass filtering, to explain how details below a threshold, in this case ~1mm, are lost in the translation from the surface of the skin to the brain. The results of this work show that Braille is more able to remain analogous to its original state than standard Roman letters. This is due the reduced complexity of Braille when compared to Roman lettering; as there is less tactile information (curves, crosses, circles etc.) to resolve, more of the information survives the cutaneous low-pass filtering effect caused by the limitations of special acuity. Although this study was done before the advent of haptically-capable touchscreen interfaces, I believe it has some relevance by showing the limitations of fingertip cutaneous sensing and may be used by designers of these types of interfaces.

Spatial acuity can easily be tested by the reader. Compare the feeling of a two-point interaction between the fingertips and wrist by touching each with a pair of pencils or compass. It is quite surprising how low the acuity of the wrist is compared to the fingertips or lips. This is again an example of necessity being the mother of invention; we interact with the world mainly through our hands and so they must be able to parse a detailed description of an object to our brains. Our wrists need only to be able to sense in a less refined way; temperature and skin deformation of any type.

Haptics in Music Performance

The effect of haptic feedback on music performance is great; musicians often refer to the “feel” of an instrument as having an effect on their performance or enjoyment of a performance. In practical terms a stiff instrument, perhaps a string instrument with too-high action (distance from the string to the fingerboard), is more difficult to play expressively than a looser one.

Musicians also use haptic feedback to navigate their instruments; playing accurately and at fast tempos requires inter-limb and digit coordination to be very accurate. The limitations of the instrument may be defined haptically, as the location of the drum plane in space or the length and width of a finger board. In these cases the haptic senses are dealing with proprioceptive information, usually referred to as muscle memory. The body has learned through repetitions how far to move the limbs or digits to achieve a given effect (pitch, articulation, rhythm) and eventually become second nature to the competent player. Cutaneous feedback is also present in some interactions; the feeling of the string under a finger is important navigate the player’s fingertip to the correct position relative to the short axis of the string to prevent mis-fingerings.

Great musicians are said to have a “wonderful touch”, meaning they are in complete control of the instrument and can physically interact with it to make wonderful music. The range of possibilities for the musician are increased as their proprioceptive familiarity and responsiveness are improved with time.

Muscle memory, and therefore kinaesthetic haptic feedback, has a great influence on the proficiency of the player. During the learning process for music students, the student must develop technique that allows them to execute actions intuitively. It has been proposed that singers employ their proprioceptive sense when aiming for pitch. Practice and repetition allows them to have a haptic image of how a pitch feels in the diaphragm so they are more able to accurately repeat it.

Gaining proficiency in a physical task involves several factors; awareness of the constraints of the task system, awareness of where the relevant appendages are within the system, and knowledge of what options there are available to move within the system. That is to say, the person knows where they are and where they can/should go next. If we imagine an improvising modern dancer, from any given position of the limbs the dancer has a repertoire of actions they can perform variations of. This is *a priori* knowledge gained through practice, study and performance, but it is executed seamlessly within the improvisation. For musicians, the process is similar; an idea is developed based on how *a priori* knowledge of i.e melody, harmony and rhythm can be applied to the current musical situation, the idea is then executed.

Speaking from my own point of view as an improvising musician, the line between imagining the idea and executing it is not altogether conscious but is also embedded in

- conscious knowledge of what the (harmonic and rhythmic) situation is
- an imagined idea based on said knowledge
- muscle memory of patterns that correspond roughly or exactly to the idea

The haptic dimension is vital, and informs the other aspects by defining the limitations of what can be done. The Jazz guitarist Howard Roberts coined the term “composing at the speed of thought” to describe improvisation, and taken to an extreme would mean that the musician would be able to “unplug” their conscious proprioceptive sense and intuitively play any idea they imagined. Perhaps the master musicians of our time have been able to do this, who knows?

To better understand the dimensionality of haptic musical interactions, so they can be deconstructed and modelled, they can be classified into three types; boundary events, surface events and vibrations.

Boundary events are typically either rigid surface interactions such as striking a drum, or tensioned spring type events like plucking a string. In these cases, movement is restrained forcefully until some release condition is satisfied, the string is plucked or the direction of the drumstick is reversed. Boundary events are usually discrete and in can ***mostly*** be interacted with in one dimension only and are ***mostly*** repeated at low frequencies (less than 25Hz).

Surface events occur in bowed instruments where stick and slip effects cause the bow to provide energy through friction to the string. This manifests itself as stickiness in this case but any event that provides continuous physical impedance to the body is surface-based. Stroked instruments and rubbed bowls are also in this category. Movement is again restrained but is typically freer at higher velocities and has no special boundaries in ***usually*** two dimensions. This type of event is much higher frequency as it describes rapidly changing small variations of force from the user to the instrument.

Vibration effects occur after the initial transient phase of excitation and describe the resonating body of the instrument. These may not be felt if the player is not directly coupled with the instrument, i.e. a drum on a stand. Vibrations by definition must be greater than 30Hz otherwise they are perceived as discrete events [Gescheider] and they are usually higher than that. The vibration frequency is a function of the excitation impulse frequency and the parameters of a resonating body.

Vibration is an example of active haptic feedback, it occurs after some other interaction and does not require any further input to actuate. Both boundary and surface events are passive and only respond to user input at that point in time.

All of these effects are interrelated and occur in almost every musical interaction, working together to provide a rich haptic experience. In stroked percussion instruments, the focus is on surface based events that are large enough to perhaps be considered as soft boundary events. This family of instruments includes the Guiro and Washboard and is present in many styles of music.

In the world of virtual instruments, great care is taken to accurately model or multi-sample high quality instruments. These are usually played with a controller keyboard which is something of a compromise for most other non-keyboard type instruments. Some work has been done to add haptic capabilities to the keyboard [Oboe] but the majority of musicians use a standard keyboard. Haptic feedback has been shown to enhance both the playing and cognition of music [Baille] [Lehmann] [Laurienti] but limitations in actuators and a lack of standardisation and support prevent mainstream music haptics from becoming prevalent.

Haptic Illusions

A haptic illusion is a stimulus is perceived as being something other than itself, that is to say extra information is added in another modality to enhance or diminish the overall experience.

One of the most widely known visual-haptic illusions is “The Rubber Hand Illusion” [Hayward] [Botvinick] which exploits an incongruity between the visually observed and haptically felt. The subject sits at a table with one hand resting on the surface. The hand is hidden from the subject and a rubber model of a hand is placed in front of them. The experimenter then begins to apply identical stimulus to both the real and rubber hands, which creates the illusion that the rubber hand is receiving the stimulus. When the experimenter removes the stimulus from one hand, the brain attempts to “fill in the gaps” by diminishing the sensation when the rubber hand is not stroked or by providing a sensation if stimulation of the real hand stops.

Charles Spence has created a body of research relating to multimodal dining experiences. A proprioceptive illusion was created together with the experimental chef Heston Blumethal, by combining a dining experience with an incongruous audio soundtrack. Diners were asked to eat soft purees of vegetables while listening to a portable music player. The soundtrack was made up of crunching noises as though recorded from inside the head. The diners reported that their jaws did not seem to work properly as their muscles received feedback the food was crunchy when there was in fact very little physical resistance in the mouth. Spence has also worked with the axis of taste and sound to modulate flavours by audio stimulus [Spence].

Haptic illusions, in the same way as optical illusions, can cause the perception of “phantom” stimuli, movement and other localisation illusions. These are termed the *phi effect* (apparent movement) and the *tau effect* (apparent distance reduction) [Hayward].

In multisensory interactions, it is typically the visual aspect that is dominant, hence the “ventriloquist” effect in an audio-visual interaction. Likewise, vision tends to dominate in haptic-visual interactions [Hecht][Mensvoort] but sensory dominance in an audio-haptic interaction is case-dependent.

The interdependency between senses, and the fact they can be fooled, is very useful for designing multimodal interfaces. It means that a simple haptic model can be enhanced with a more complex audio model and vice versa, and that a lot of the time the brain provides information based on what it expects to experience, rather than what is actually happening. However, tight synchronicity is a requirement and responsiveness in musical interactions is of paramount importance.

Previous work in Haptic Instruments

There are two broad categories into which haptic instruments can be divided; emulations of existing instruments and original instruments. The term "original instruments" cannot have an absolutely concrete definition, as a lot of this work is very much informed by traditional music instruments. Where these instruments differ from the emulations category is in that they allow for interactions not available or not possible on an existing instrument.

Original Instruments

Some original instruments attempt to improve the performance capabilities of the musician, either by facilitating performance articulations difficult or impossible in real life or by working to train the musician.

Edgar Berdahls Haptic Drum [Berdahl] is an example of an original instrument that is based on an existing instrument and therefore an existing interaction methodology. The device is a snare drum with a speaker mounted inside the shell, between the heads. When the drum is struck, a delayed sample of the strike is played back from the speaker which causes the skin to deform, starting in the opposite direction of a normal strike and thereby pushing the stick back up. This allows the player to perform one handed drum rolls by applying downward pressure onto the skin with a drumstick. Varying the pressure alters the dynamics of the sound and response of the instrument. This instrument extends the capabilities of the musician, and those with better technique will get more out of it by mixing traditional playing articulations with the new ones afforded by the Haptic Drum.

Similarly, The Overtone Fiddle [Overholt] allows new articulation on an existing musical platform. A traditional violin is reworked with actuators to allow the body to resonate at a given frequency which causes sympathetic ringing from the strings. Many new articulations are possible with additional control mechanisms while the instrument retains its traditional sound and interface.

Bongers experiment with the LaserBass [Bongers 1998] is an example of extending the Theremin concept with a tactile aspect. The player plucks the virtual string, an invisible beam of light, and the position of the other hand determines pitch in a model of an upright bass. A "Tactor" worn on the fingertip provides tactile feedback to the player but the study concludes both that the haptic sensation is too weak and that a small delay in the respective onsets of the audio and haptic events spoils the coherence of the experience. A faster and stronger device, the Tactile Ring, performs better and allows a more natural interaction and demonstrates the need for a fast actuator response time.

Emulated Instruments

Yuan's "Blind Hero" adapts the "Guitar Hero" concept for the sight-impaired by using haptic signals to display the next step in the sequence. The physical interface is a small toy-like guitar with five buttons on the neck and a haptic glove with one motor fastened to the back of each finger. Here the haptics function only as a source of explicit information; where the next step is and its length. There is no haptic display of musical information in this study. Although the authors show that a Guitar Hero type game is made accessible to the visually impaired, the dimensionality of the haptic display does not allow a 1:1 port of the system from the standard version, i.e. it is only possible to "see" one step ahead. The authors mention how the haptic system functions as a memory aid when practising the same task multiple times; the user learns the order of the sequence through the haptic display and commits it to memory. This is typical of muscle memory tasks in music performance. Using standard eccentric motors found in pagers, the authors mention the challenge in rendering short and long pulses at equal amplitude. These type of motors have a significant ramp up/down time and so rendering short high amplitude signals is often not possible.

The Haptic Drum Kit [Holland] is an “open air” instrument that functions as a trainer for drummers to learn limb independence and polyrhythmic structures. The idea behind this study is based on work by the music educator Emil Dalcroze who proposed that complex rhythmic structures are easier to understand, retain and repeat if the musician feels the sequence rather than hearing or reading it. Dalcroze proposed that his students enact rhythms with their bodies and limbs; cutaneously by tapping fingers, hands and feet and propreoceptively by waving the arms and head in space. The Haptic Drum kit consists of four vibrotactile motors attached to the wrists and ankles and an Arduino and Max/MSP control system. The system has a maximum 50% duty cycle pulse frequency of 10Hz and the motors require 50ms to come to a full stop.

Five drummers played a MIDI drum kit in an experiment to match straight, clave-based and polyrhythmic patterns presented as audio, audio-haptic and haptic. All the participants performed worst with only haptic feedback and the author notes the “quietness” of the motors as an important element. An interesting point is that the haptic signals also contain the information of which limb should play what drum, and the drummers learned the pattern sooner in the polyrhythmic exercises when the haptic signal was present. The next stage of this study is to implement Tactor devices in place of the motors as they are higher amplitude and with a wider dynamic range.

The HaCHIstick [Hachisu] is a beater-like device with an embedded actuator designed for use with a custom tablet application. The player strikes the beater on the surface of the tablet, displaying a glockenspiel GUI, and the voice-coil actuator simulates the haptic response based on the velocity and position of the beater and the material that the virtual instrument is made of. The target motion for the simulation is striking, and the authors use soft rubber, both on the head of the beater and as a transparent sheet over the tablet, to eliminate as much of the naturally occurring vibrotactile feedback as possible. This allows them to render a variety of effects without the “normal” haptic feedback confounding the interaction. However the viscosity of these materials causes a problem when attempting lateral movements (i.e. glissandi, trills etc.) as the two rubber surface have some amount of inherent stickiness. Ideally, the player would want to feel the gaps in the tines as haptic events and the authors state that they plan to iterate on this design by experimenting with different materials.

The GUI///RO [Müller] is a stylus and screen based haptic interface for music. The authors present a touch sensitive and accurate proprioceptive haptic stylus which employs an electromagnet and steel ball as the haptic actuator. The voltage across the magnet is manipulated according to the force model in the system and can provide a wide range of physical resistance as the ball causes the stylus to “brake.” The touch screen interface allows accurate tracking of position and velocity and a force sensor in the stylus allows force sensing in the normal direction. The authors do not divulge any performance data for the system, although it is based on Arduino and certainly adequate.

Real instruments can have their physical properties manipulated using haptic feedback. The Force Feedback Keyboard [Oboe 2006] can provide the sensation of resistance in a keyboard instrument. Each key is equipped with sensors to measure the position and speed and coupled to a weight mounted on a motor. The system can manipulate the weight of each key and give the impression of stiffness or heaviness to a player. The study uses Spring-Mass-Dampener (SMD) models to emulate the responses of various keyboard instruments; Grand Piano, Hammond Organ and Harpsichord. The player's input, the pressing of the key, is applied to the SMD model and the result rendered haptically using the weight system.

The system can also render inertia and gravity effects, for example the heaviness of the Grand Piano keys causes some free-fall if tapped quickly, whereas the sprung Hammond Organ keys spring back to position. The software used in the system renders a simplified Impact-Decay version of the SMD model, as the full simulation is too computationally costly. However, a lower-fidelity version is still able to provide a realistic experience, perhaps also due to the consistent visual stimulus – the input device is a section from a real organ keyboard. This system is also complex in its implementation, with proprietary hardware (the "circular" motor) and the mathematical modelling of real physical events. Important to note is that this study is one of degrees of realism – the visual and cutaneous aspects of playing a keyboard are already present so any small discrepancies in the haptic modelling may not be noticeable.

Modelled Haptics

Many different hardware systems have been employed in research and commercial products to simulate friction and other haptic events.

It is important to note that the model is always a compromised version of the real system. The modelled system must be reduced in complexity in order to be manageable and the degree of "perceptual complexity" describes when the level of detail in system is too small to be perceived. In most cases, a great deal of low-level detail can be excluded from the model.

In the mid 1990's Sony used motors in their "Rumble Pak" controllers to render various effects. Although the resolution of the motors was limited the congruency with the audiovisual elements made the friction effects in, for example, racing and fighting games an enhancement to the experience. Some music based games (i.e. "Dance Dance Revolution"," Wee Papa the Rapper") used the motors to render rhythmic pulses, musical phrases and silent error messages. Computer users, gamers in particular, have been experiencing haptic friction for some years and many high-end joysticks have built in haptic capabilities.

Force-feedback devices can simulate a wide range of haptic experiences and also render objects in a one-sided three dimensional space. Typical of these is the PHANToM device by SenseAble Technologies which consists of a base unit and a stylus attached to the base by three jointed arms which allows the stylus limited freedom in 3 DOF. Chen et.al uses this device to model various states of friction on a virtual surface by modelling the attachment and release phase of a stylus entering a dip on a surface. This work presents a simple model of adhesion similar to a Mass-Spring system and a breakaway model that depends on the normal force of the stylus. This model is successful at rendering friction effects with low dynamic range, i.e. two almost-smooth surfaces in contact under normal and lateral force, but no mention is made of how it performs at modelling more distressed surfaces. In a summary of an earlier prototype, the authors remark on their method of calculating velocity as a derivative of position, "*Noise and discretization in these measurements cause this velocity calculation to be unstable when the stylus is mostly still*" [Chen]. The later prototype measured the velocity using the force sensors built into the device and resulted in an improved performance.

[Salsbury] defines two types of haptic interactions; those caused by an encountered objects' geometry and those caused by its surface properties. As an example in a one DOF system, a haptic wall is rendered when the input crosses an arbitrary *x*, by increasing the force to maximum and maintaining it until force in the opposite direction is measured. By varying force feedback along one DOF, as a function of length, displacement effects can be rendered [Snibbe]. Snibbes' model of this effect is shown in the diagram below (from [Snibbe]). The arrows represent the magnitude and direction of the force applied. The effect is that the top surface is perceived as something like the bottom one.

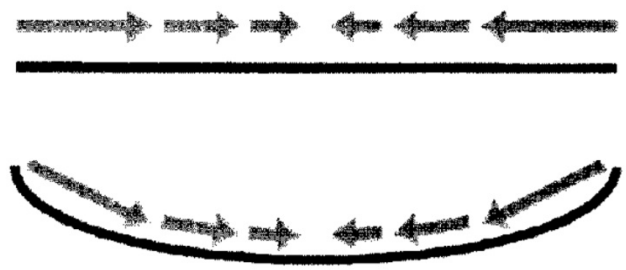


Figure 2 Snibbes' model of rendering a dip in 1 DOF

This is similar to "Force Shading" [Morgenbesser], which describes a surface by varying the force output to the user as they traverse a surface; "*creating the illusion of a non-flat shape on a nominally flat surface*" [Morgenbesser]. Force shading combines software modelling with haptic illusions and exploits the limitations of haptic perception. It uses polygonal meshes instead of mathematical formulae to describe surfaces and interpolates between the points in the mesh using Force Shading. The result is an effective rendering system with low complexity, which in turn means that complex effects can be rendered robustly and easily.

[Morgenbesser] presents a simple model of rendering a cylinder using a vector-based approach by calculating the force as a function of distance to the central point of the cylinder. This approach has disadvantages when rendering impenetrable surfaces and is compromised when force is a position derivative. However, for a 1 DOF interaction a simple model would be suitable as much of the drawbacks are relevant to 2 DOF interactions. Force Shading can be thought of as the haptic equivalent of a Bump Map in 3D graphics; the z axis is set to a point z = f(x,y) and interpolated to all neighbouring z. The author performs classification and matching experiments using a PHANToM device and the results show how shaded polygonal features and high resolution renderings are perceived as equally smooth .

Snibbes terms the marking of content in the haptic domain as "Haptic Annotation". This is particularly relevant to musicians in terms of silent cues, usually read from a page or signalled by a conductor, which can be provided without disrupting the performance, as in [Holland]. An example of this is to use normal force to increase the level of detail when browsing a 1 DOF haptic interface.

TeslaTouch [Bau 2010] uses the concept of "electrovibration" to alter the friction between a moving finger and the touch-sensitive panel surface. By altering the frequency and amplitude of a periodic voltage, the system can provide the sensations of stickiness, waxiness and bumpiness among others. As there is no physical actuation taking place in this system, the haptic sensation can remain absolutely consistent over the entire surface which is not possible using actuator arrays. The TeslaTouch is a multi-touch device and uses computer vision for tracking of the users' fingertips [Bau 2010]. The experiments presented in this study show how low (80Hz) and high (400Hz) frequency AC stimulation cause different perceptions dependent on the respective amplitudes at which the stimulation is delivered. They also show that the Just Noticeable Difference thresholds for frequency and amplitude are about the same for other lateral actuation. This technology offers several advantages over traditional moving-part actuation; it is silent, it will scale uniformly and it cannot wear down. However, the primary disadvantage is that it will only work on moving fingers so therefore it can only simulate specific interactions, and the hardware is considerably more expensive than mechanical actuators. The user also must be electrically connected to ground and the system has high power requirements.

Most researchers emphasise the need for a responsive system over a detailed one. In the haptic domain, particularly with stimuli in other modalities present, temporal congruency is important. Physical models of the systems, based on force functions, permit a range of effects to be rendered and scope for alteration.

Controller Instruments

Controllers in academic work are typically designed to address a subset of features or behaviours of the controller. Accuracy and quality are usually emphasised over price, portability or ease of use. Cadoz' Modular Keyboard [Cadoz 1990] illustrates this by employing large motors to change the inertia, mass, and response to gravity of a key. The system is bulky and expensive but offers a highly realistic simulation of various keyboard types.

Many computer-based musicians are composing, performing and recording entirely in an integrated workstation environment and using external devices as input controllers. These are mostly emulations of existing instruments such as piano keyboards, drum kits and saxophones.

Modern recording technology offers the musician a vast array of sounds and an almost infinite scope for processing them. Sound can be recorded into the system as audio via an A/D converter and can also be triggered by events sent from a device to a "virtual instrument" software plug-in. For the average budget- and space-conscious computer musician, a trade off must be made between the number of instruments available and the means by which to control it.

The MIDI protocol has been standardised for over 30 years in computer music and allows for the transfer of data at high speeds to and from a controller and host. The keyboard controller is the most common MIDI input device and is familiar to the majority of musicians. Other MIDI controllers have been made to accommodate the playing styles of other instruments, for example the Akai WX Saxophone breath controller and Roland V-Drums drumkit.

The keyboard works well for most boundary events, which include plucked strings and struck percussion, but it has no real mechanism to allow continuously varying input. This means that surface events cannot be controlled adequately. Some modulation of continuous events is possible, for example using an expression wheel to add vibrato to a held violin note, but it is physically disconnected from the action that sustains the sound, and does not resemble the action of applying vibrato. In contrast, striking a key and hearing a drum sound is less incongruous as the action and outcome are related in a familiar way.

Figure 3 MIDI controllers. (Left to right) Portable keyboard with continuous encoders, breath controller and Buchla Thunder.

Don Buchla's range of synthesiser controllers encompasses both open-air controllers ("Lightning Rods") and surface-based interfaces. One such example is the Thunder controller which is a multi-touch, pressure sensitive device for music performance. Like many of Buchlas' devices, the controller is highly specialised and requires a significant adaption of playing technique to perform with well. However, the surface can be used to record gestures and map them to appropriate sounds. [Pinch 2004]

In spite of this, bowed string instrument models, designed to be used with a controller keyboard now incorporate features such as key-switching to change articulation, and can be played highly expressively. This has had the effect of increasing the number of soundtrack composers and reducing the demand for studio session musicians. High-dimension controllers such as large drum kits are less popular, as are controllers with a limited range of applications like Buchla's range of controllers. This is perhaps due to their cost, size or lack of adaptability to other scenarios; you can play drums on a keyboard but the opposite is not true. Musicians must compromise based on the requirements of their situation.

Some of the most important controller preferences and needs of the average computer musician are;

- Familiarity
- Size and Portability
- Cost
- Adaptability

The keyboard fills these requirements for most boundary and some surface event musical interactions but stroke-able percussion instruments cannot work on a keyboard as they rely on continuous input in two dimensions with directionality. A controller that allows continuous input, such as a breath controller, presents an obvious incongruity, and the keyboard cannot parse the velocity data required to generate a rhythm. When and if these instruments are available to the computer musician, they are in my experience trigger-able samples with some tempo locking facility.

With the needs of musicians in mind, I will explore the potential of adapting a computer mouse for use as an original haptic instrument and controller. And take inspiration from [Cook], "existing instruments suggest new controllers."

The following questions summarise what I will investigate;

- Is a haptic mouse feasible as a music controller?
    - Technologically?
    - As a mode of physical interaction?
    - As an analogue of the real instrument?
- Can the haptic mouse offer a "better" experience than other controllers?
    - Can it offer more control over the interaction?
    - and therefore more potential for expression?
- Can the haptic mouse be adapted to other interactions?
    - Can soft boundary events such as strings be controlled convincingly?
    - Can a rigid wall drum simulation offer any advantage over a keyboard?

- What is the potential of the platform?

Terminology

Throughout this paper, I will use various terms relating to music, electronics and other subjects that are best clarified once and then may be referred to by the reader, and so as to avoid repetition and unnecessary repetition. Many of these terms have synonyms, particularly in music notation, and I have employed those most familiar to myself as a western non-classical musician.

Music

Note ***subdivisions*** are how many times a bar is divided into ***beats***; the higher the number of subdivisions the more beats there are in a given bar. A bar is a unit of musical time consisting of four quarter notes, eight eighth notes or sixteen sixteenth notes. A normal way to count these subdivisions is as follows;

- (Quarter) 1, 2, 3, 4
- (Eighth) 1 and, 2 and, 3 and, 4 and
- (Sixteenth) 1 ee and ah, 2 ee and ah , 3 ee and ah , 4 ee and ah

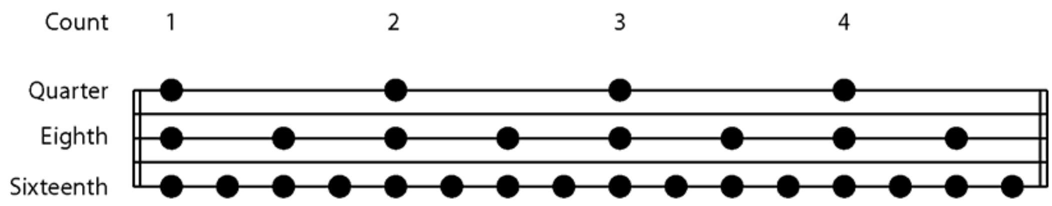


Figure 4 Notation of rhythmic subdivisions

There are other compound subdivisions, most commonly triplets (1 and ah, 2 and ah...) in various configurations i.e. six or nine, but it is most important to see how the number of discrete events doubles with each increasing regular subdivision; four, eight and sixteen.

Tempo is the speed at which beats are played in succession and is measured in beats per minute, most popular music would fall in between 60 and 180 b.p.m. This characterises how fast a piece of music is played.

Musical Dynamics are one of the five elements of music (rhythm, dynamics, harmony, melody, timbre) and generally refer to how “loud” an element is. However, a piece that is said to be musically dynamic will have a balance between strong and not-so-strong elements, both over time and in terms of the instrumental arrangement.

Timbre is the tone of an instrument, a harpsichord could be said to have a thinner and brighter timbre than a grand piano.

MIDI stands for Musical Instrument Digital Interface and is a standard for the transfer of data between devices. MIDI signals from a controller keyboard or drum pad consist of a note-on message for a given pitch and a note-off. Additionally MIDI supports 8 bit Control Channel data for continuous encoders such as knobs, expression pedals and wheels.

Electronics

Voltage is the electrical potential between two points, and is measured in volts. Direct Current voltage (DC) is steady state and constant whereas Alternating Current voltage (AC) changes with time. All audio signals in the electronic domain are AC signals as it is the change in voltage that allows the sound to manifest, eventually by modulating a speaker cone.

Current is the flow of electrons across a point and is measured in Amperes, or Amps. Current is the lifeblood of the circuit and the flow of current around the circuit, according to predefined conditions, is what makes it work.

Electrical Resistance is the property of a body to impede the current flow through it and is measured in Ohms. *Resistors* are passive components that provide a predetermined resistance and may be fixed or variable. Most analogue devices with knobs use a variable resistor, or *potentiometer,* to control important parameters, i.e. the volume on a stereo system.

Design

This section details several prototyping and analysis sessions based on conclusions from earlier work and current research. The section concludes with a design proposal for a final prototype.

Considerations

Having already defined some initial criteria for the capabilities of the controller in a general way, considering it in context will provide some inspiration of initial design work.

The simplest interaction would perhaps be dragging a stylus along a surface by moving the mouse from side to side. If we break down the real interaction, the parameters that make up the interaction are;

- Varying physical resistance to lateral movement
- Cutaneous impacts as a function of lateral velocity
- Synchronised impact sound

A model that emulates these three characteristics will at least be suitable for evaluating the feasibility of the system.

Earlier Design Work

In my previous work, I concluded that a computer mouse, extended with haptic capabilities, would be the ideal input device for the simulation. It affords 2 degrees of freedom for motion and the standard set of actions for playing stroke-able percussion can be almost directly translated. The mouse accommodated the natural arc of the players arm and can house additional sensors and buttons for normal force or other input.

The mouse is placed on an iron mat which allows a mouse-mounted electromagnet to adhere to it when a condition is met. My previous system rendered a sinusoidal surface by varying the grip of the magnet as a function of the player’s position, velocity and normal force inputs.

This work continues this previous work by refining some its concepts, prototyping and developing a new system and by performing experimental work.

Interaction Requirements

At the cognitive level, the user should be thinking as a musician and so the interaction must be musical in some way. The rhythmic stroking motion of playing the instrument is what defines the tempo and each stroke is typically subdivided by at least 16^{th} notes. The speed and range of this motion should be analogous to a real-world instrument of this type. In order to allow better control over the simulation, the interaction must be responsive to rapid and small changes. The mouse is a familiar device, and the user has *a priori* knowledge of how to manipulate it, so maintaining as much of the standardisations as possible is important. This could be for example, hand position, the button interface and the location of sensors.

The traditional gestures from the real instrument should then be mapped to the virtual. The stroking motion is similar with both mouse and stylus although the position of the fingers is different but the mapping is close to 1:1. Other articulations, such as shaking or tapping the surface will be discussed later.

Hardware Requirements

The most important hardware requirement is that it not infringe on the range and speed of motion of the mouse. There will be natural haptic resistance in the system, as gravity pulls the mouse onto the mat and impedes the lateral motion. It is important to make sure that this resistance is as uniform as possible so that the simulation may run unadulterated.

With any vibrating haptic device there will be audible noise during actuation which must be kept to a minimum. To allow the maximum range of effects, the system should be able to render stiff walls. This requires that the magnets are strong enough to do this and that there is a mechanism for deriving force independently of velocity [Salcudean].

Software Requirements

The software must consist of an audio and haptic model of a percussion instrument and a front-end GUI application for altering the parameters of the instrument. The haptic model must be able to render surface events at high resolution and minimal latency. The audio and haptic output must be synchronised consistently to under ~10ms. The model must be flexible enough to vary the instrument to produce different sounds and haptic effects.

The level of detail should be such that a convincing emulation is rendered; this may mean one of the sensory outputs may be of lower quality than the other. The software should be able to map the user input to the audio-haptic output as accurately as possible in both time and space.

In previous simulation designs, I have used a fixed model of a surface and mapped the users' derived position to it, which required quantisation and error checking, and proved to be sometimes ineffective. The mapping of the model to the user's input must be as close to 1:1 as possible, to avoid the surface feeling like it is changing size or position.

Prototyping

From previous experience, I have learned that the design of the system on paper can often be difficult to implement exactly or that unforeseen issues can arise once a prototype is built. This is especially true when a system has physically moving parts that are only partially constrained or have a small tolerance for positioning or coupling. Problems such as quantisation errors or physical discomfort are also only able to be evaluated on a working prototype. To this end, the prototyping process will be iterative and documented from the very basic stages to the final design.

Requirements

The nature of the simulation demands proprioceptive haptic feedback; the hand must be impeded to simulate the change between a dip and a bump. Additionally, solid wall simulations are very useful in a percussion instrument, both as strike-able surfaces and as end-stop indicators. To this end, the actuators must be strong enough and responsive enough to fully impede the movement of the mouse with no noticeable deceleration. The actuators must also be capable of rendering low-level impedance, as found in a lightly ridged washboard and perhaps deliver cutaneous feedback to simulate impacts. The mouse must be as easy to hold as a standard mouse and be unobtrusive to the interaction.

Cost

Bearing in mind that this system should be in some way practical to build or buy, the cost must be kept low. High quality haptic devices; Phantom Omni or Braille readers, are expensive and are certainly luxuries for the average haptics-curious musician. The nature of prototyping also demands that protypes are discarded as part of the iterative process so the initial costs must be limited if a final prototype is to be built.

Magnets

Two main types of magnet, which are small and strong enough to meet these requirements, are currently in mass production; standard electromagnets and bipolar electromagnets. Both work on the same principle; an electrical current is passed through a coiled wire which generates a magnetic field around an iron core. The direction of the current from one end of the coil to the other determines the polarity of the field. The strength of the magnet is determined by the number of turns in the wire and the mass of the iron core, therefore stronger magnets are larger and heavier than weaker ones. Bipolar magnets are stronger than a same-sized electromagnet and the field has a wider reach.

Figure 5. Various sizes of small 12-20 volt electromagnet

A standard electromagnet can typically only attract an object when it is very close whereas a bipolar magnet pulls approximately 30% of its maximum lifting force at 3mm. Standard magnets are cheaper than bipolar, have lower power requirements and available in a wider range of shapes and sizes. The magnets strength is expressed as lifted kilograms; that is, how much weight the magnet can suspend reliably. The actual maximum lift may be more and also depends on the supplied voltage. This is a useful relationship as it allows the strength of the magnet to be controlled by a fluctuating signal.

Useful Components

Additional haptic actuators available to the prototyping session are motors and solenoids of varying sizes and strength which can be used to render cutaneous feedback. Buttons and force sensitive resistors are available as input devices. The H-Bridge chip can invert the polarity of the current with a 2-input logic gate (OFF, +, -, OFF). This allows the current to be reversed immediately before turning off to dissipate the remaining magnetic field. This has the effect of causing a motor to brake and also allows the behaviour of the magnet to be “sharper.”

Prototyping #1

The purpose of this prototyping session is to determine the strength of the magnets and evaluate the surface material.

Two types of electromagnet are selected; a (Ø) 12mm * 20mm (h) that can lift 10kg and a larger 30mm * 20mm that can lift 25kg. Both magnets run on 12vDC, the smaller one draws around 330mA of current and the larger 660mA. A 900mA 12vDC supply is used with a simple breadboard mounted switch to apply the current to the connected magnet; this allows very rough “handmade” surface effects to be rendered, albeit at a binary resolution.

The surface is a 400mm * 250mm iron plate, 0.5mm thick. It is lightweight and smooth, but not polished, on the surface.

Applying full current to both magnets causes them to adhere to the surface so they cannot be moved; the small magnet can be pulled off by applying an impractically large, for this work, amount of force. The smaller magnet is therefore selected as being adequately strong and is better to handle than the bulky 30mm magnet. As the surface and underside of the magnet are not perfectly smooth, there is some resistance between the two when dragging. Sanding and buffing both surfaces helped to alleviate this but the effect is still quite different from dragging the magnet on a neoprene mouse mat surface.

Raising the magnet away from the surface by some mediating layer reduces the maximum grip the magnet can produce, as a function of the thickness of this layer. For this magnet a distance of ~2mm will prevent the magnet from adhering. It is important that the layer is smooth, even and as thin as possible. To accommodate the slight imperfections in the surface, it makes sense to use a deformable material for the layer. The layer may also be attached to the magnet itself or cover the entire surface. Several methods were prototyped;

- Disposable 2-ply tissue on magnet
- Microfiber cloth on magnet

- Tape on magnet (Gaffa and Sellotape)
- Paper on surface (1,2,3 and 5 A4 sheets stacked)
- Nylon tights material on magnet (polyamide/spandex mix)
- Polyester material (imitation silk) on magnet

Figure 6 Large magnet (left) and two smaller magnets with fabric covering.

The disposable tissue allowed maximum grip but became quickly torn from stresses going into a full stop. This caused gaps in the tissue to affect the feeling of the magnet-surface interaction depending on direction.

The microfiber cloth allowed a very smooth travel across the surface but reduced the maximum grip substantially making it easy to break from the surface when full current was applied.

The tape caused stick-and -slip effects that interfered with the movement, which were especially noticeable when applying downward force. The tape also quickly deformed to the shape of the magnet and provided no buffering from the surface.

The paper became damaged in a similar way to the tissue, although to a lesser degree. The paper was also not deformable enough and still caused some scratching sensations at certain magnet orientations. Using 3 or more sheets of paper reduced the maximum grip and allowed relatively easy breakaway.

The nylon material allowed smooth travel and maximum grip but tended to "bunch up" and gather under the magnet. Securing the cloth with a cable tie around the magnet solved this problem for the short term but some adjustments had to be made after using it for a time. This material is porous and also developed some holes after around 5 minutes of continual use.

The polyester allowed maximum grip, was very smooth and allowed unhindered movement. This material is much stronger than the other types that also allowed maximum grip and can easily be fixed around the base of the magnet and held in place with a cable tie. There may be a slight benefit in covering both the surface and the magnet with this material but stretching it across the mat surface may prove difficult.

Occasionally, the magnet remains magnetised for a short period after the switch is disengaged. This may be residual current in the coil causing the magnet to remain active in which case an H-Bridge implementation could perhaps solve this. It must also be noted that the maximum grip is higher than what is practical, in this case the rendering is essentially a solid wall and the required voltage to render this may actually be lower. (There will not be a true DC signal going to the magnets but a max 99% duty cycle PWM AC signal).

Prototyping #2

In this prototyping session, I compared different types of control signal (see Control Signals) and worked on parameter scaling and some basic haptic effects using the magnets. I also experimented with deriving direction using two buttons.

The purpose of the buttons is twofold. When rendering stiff walls, it is important to be able to derive force independently of motion as discussed in [Salcudean]. The buttons allow the user to apply some constant force to the model in either direction by pressing them - thus allowing for an escape from a solid wall. They also permit additional features to be explored - buttons can be pressed independently or together to alter the surface in real time. The location of the buttons [Photo] is such that the user intuitively presses them when moving the mouse in the opposite direction; left button when moving right and vice versa. Given a solid wall on the right, blocking further movement, a press of the right button could release the magnet and then test for movement in the opposite direction.

Prototyping #3

The purpose of this prototyping session is to work with the magnets on the surface finding the best position for them and to integrate the mouse, actuators and sensors in a way that is comfortable and will allow ease of use.

The magnets and mouse both have a very low tolerance for functioning above the surface and must therefore always be as tightly coupled to the surface as possible. Above 2mm, the magnets will not attract the surface and the mouse tracking does not work at all. The magnet must then be mounted in one of two ways; inside the mouse, adjacent to the mouse IR sensor used for tracking position or adjacent to the outside of the mouse and held in position. Opening the mouse shows limited space for any additional components due to the circuit board onto which the IR sensor chip is mounted. Moving or damaging the chip would destroy the mouse so a smaller travel mouse was used for this prototype.

Figure 7 Prototype using a travel mouse and magnets

Holding the magnet in place, I moved it around the plate activating it with a periodic signal. When holding the magnet on the side of the mouse, the opposite side of the mouse sometimes rode up as the magnet engaged on the other side. At high frequencies, this caused the mouse to rattle as it jumped around on the surface. Holding the magnet at the front of the mouse caused the mouse to rotate slightly when encountering larger haptic events, around the axis of the magnet. This was distracting and tiring on the wrist to correct. Holding two magnets on either side of the mouse stopped the rattling but was awkward to hold as the thumb and middle finger rested on the outside of the magnet. Putting the two magnets side by side in front of the mouse left the mouse feeling top-heavy, but worked as the best compromise by eliminating the other issues.

The mouse and magnets were mounted onto a plastic base, cut to fit each component, with wires extending away from the button end of the mouse. A small Plexiglas platform was bolted on to the magnets, holding them together, and to mount the force sensitive resistor onto in the final prototype. This allowed the mouse and magnets to be moved together at reasonable comfort but the weight of the magnets meant that the assembly was easier to use if the hand was moved forward so the index finger rested on the plate and the thumb and index finger on each magnet .

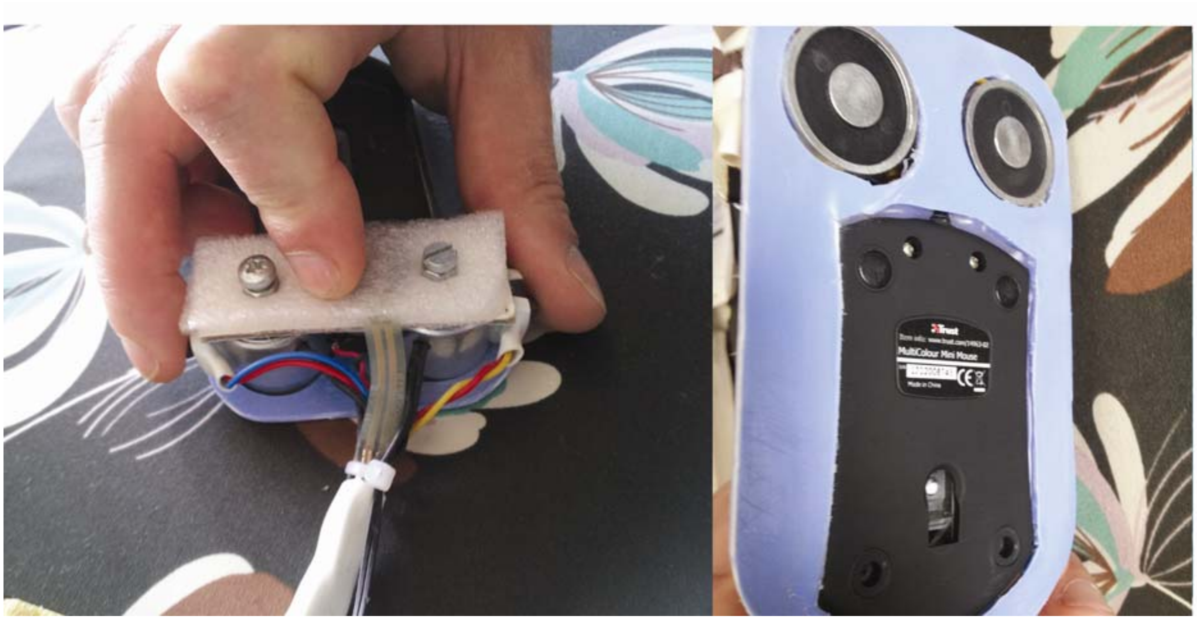


Figure 8. Prototype sensor assembly and magnet mounting

As there is a requirement for two buttons to signify direction, these can be mounted onto the magnets to ensure they are always fingered. I tested with two unconnected buttons and found that pressing the buttons when moving the mouse is almost a matter of course, as it is the thumb and middle fingers that are exerting the lateral force on the mouse. This is very beneficial to the final implementation as it will allow for a better solid wall rendering and could also be used for additional parameter control whilst maintaining user comfort.

Implementation

Mouse Data

The ergonomic comfort and tactile familiarity are the main reasons for using the mouse as an input device, but there are a few limitations in the way the mouse interfaces with the host computer.

Firstly, the data capture rate for the mouse is fixed by the operating system (Windows XP SP3) at 120Hz and then down-sampled to 60Hz into the Processing sketch which captures the mouse position and button states. The latency is low enough to be unnoticeable and in earlier experiments, I have observed that responsiveness is of paramount importance to an enjoyable and believable haptic experience.

The second is issue is that in terms of the physical world, the mouse velocity affects the sampled position. That is to say that moving the mouse 10cm on the surface will not always result in a 10cm increase in position in the model; as the graph below shows, there is some dependency on velocity. The “Fast” trace is offset for clarity.

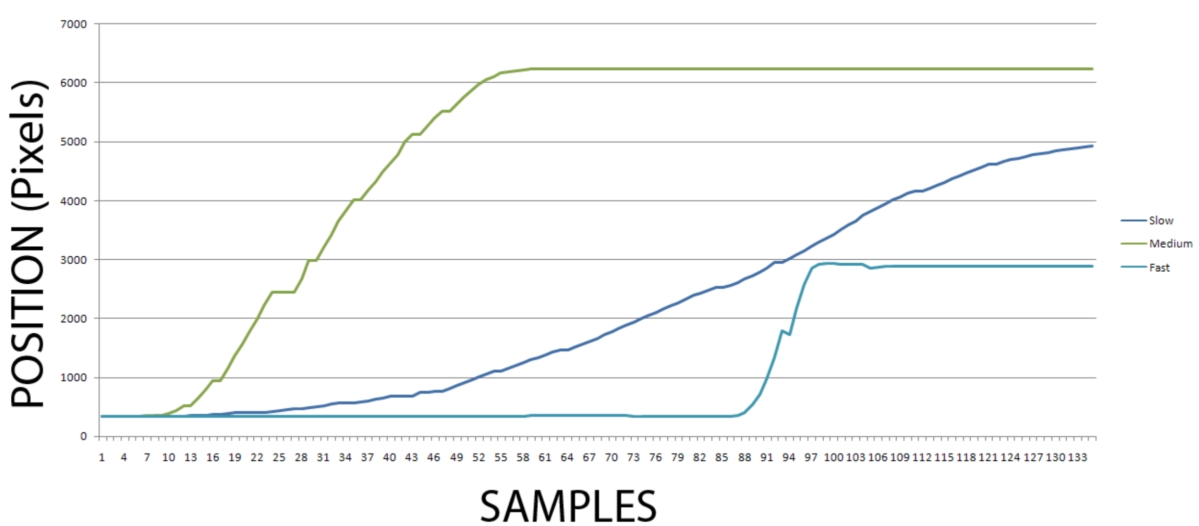


Figure 9 Graph of sampled mouse position for three stoking speeds over the same physical distance (20 cm).

[Figure 9] shows the data captured from the mouse when it was moved by 20 centimetres at three different speeds. The mouse was blocked by a solid surface on each side so all physical distances are the same. Care was taken to only move the mouse on its X axis. The speeds were gauged by hand as being from the slowest to about the fastest one would stroke a Guiro, and an arbitrary intermediate stroke was also recorded. If one assumes that the slowest stroke, and therefore the one represented by the largest number of samples, is the most accurate, then faster stroking speeds cause a larger error than medium speeds. This was true for 3 similar trials which each showed, as the graphed data does, that a medium stroking speed results in the largest reported distance. This is perhaps unexpected; it would make more sense that the slow speed is most accurate so there may be some aliasing from the CMOS sensor used to gather the position data if its response is non-linear in relation to mouse velocity. In any case, it is clear that the “analogue” velocity of the user pushing the mouse affects accuracy of the position data.

Assume that a reasonably fast stroking speed for a Guiro would be 16th notes at 120 beats per minute, or one stroke in one direction every 125ms. This is quite fast in practice, and represents the upper limit of stroking speed as it would be a very busy Guiro part to listen to and is quite challenging to keep up for any length of time. Ideally, the position variable should oscillate between two points as the mouse is dragged back and forth. This is not the case and is, again, dependent on the mouse velocity.

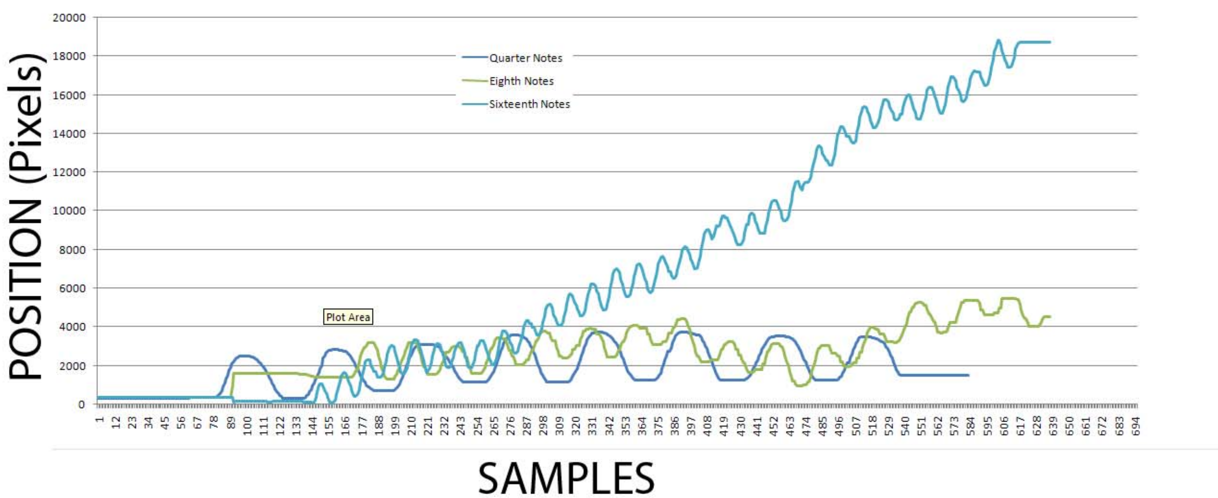


Figure 10 Graph of sampled position for rhythmic stroking at three subdivisions of 120 bpm.

[Figure 10] shows how position changes as the rate of the back-and-forth stroke changes. The mouse was dragged back and forth across a ~7cm area in time to a Smartphone metronome set at 120 beats per minute. A quarter note stroke is pushed forward ***or*** backward once per beat (counted as 1, 2, 3, 4 and every 500ms), an eighth twice per beat (1-and, 2-and, 3-and, 4-and, every 250ms) and a sixteenth four times (1 ee-and-ah,2 ..., every 125ms) so for each subdivision, the speed of the stylus should double if the player is keeping time. The graph shows a clear error with the rapid sixteenth note stroking, some amount of ‘extra’ position data accumulates as the fast speed is maintained. Mapped linearly to a model, this would result in the stylus moving over the surface laterally while the player is stroking back and forth. The lower speeds also exhibit some offset, but considerably less than the fastest. It should be noted that the offset is positive which may be due to handedness; I generated the data and am right-handed, and the position is measured as negative to positive from left to right. It may be interesting to find a left-handed subject to test this but when I perform the same movements left-handed, the result is the same [Figure 11].

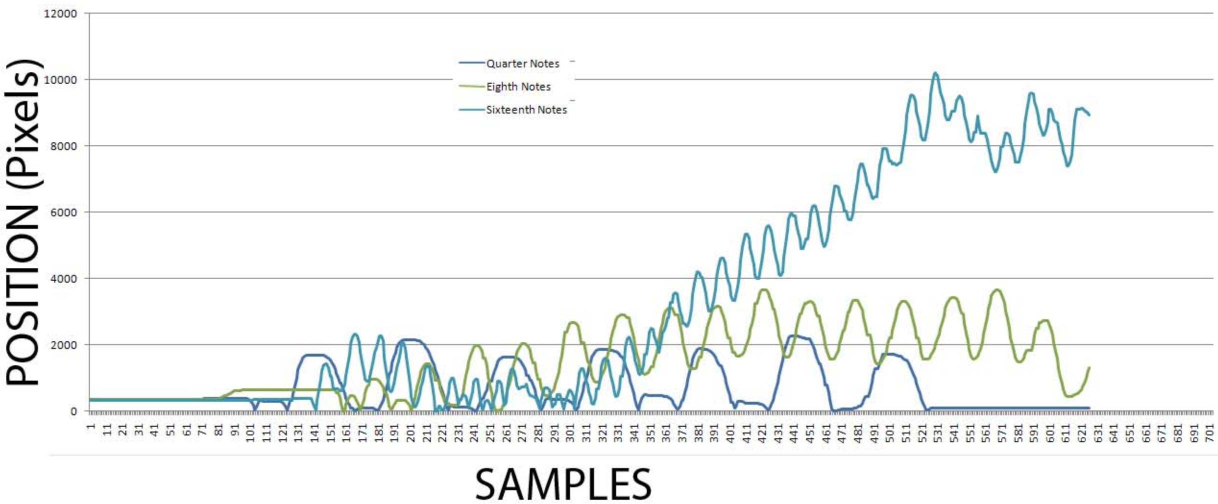


Figure 11 Graph of sampled position for left-handed rhythmic stroking at three subdivisions of 120 bpm.

The velocity-dependent position offset would certainly be problematic for models that require a wide spatial dimension in which to operate. Given a sparsely-populated haptic simulation i.e. a harp, the strings would appear to move as the offset accumulated and the haptics become rendered in the wrong position. This would make it impossible to locate specific strings and make general navigation difficult.

In the case of a guiro or washboard, the surface is periodically stressed and so can be represented by a repeating function [See Surface Function]. As the function operates on a subset of the available space (some {position, position + n} range), the position can be quantised to this range and the offset managed, as it will have less time to accumulate before being reset. So after one bump is passed the position is reset to 0. This range could be very small, a typical guiro has one raised bump every centimetre.

Interpolation

In the Processing sketch, the position is capped between two pixel values and this variable is used to display the virtual stylus onscreen. The sampled position value is reset when the user presses one of the side buttons, thereby starting the simulation. The acceleration of the virtual stylus is an influential variable in the Surface Function, as it defines the spatial distribution of haptic events i.e. as a derivative of position over time.

From position, ***p***, velocity, ***v***, can be derived as an average of the last two positions, distance travelled over a single time step. One time step is 16.6ms, sampled by Processing at 60 Hz and then converted to an audio signal in Pd, which equals a ***t*** of 735 samples.

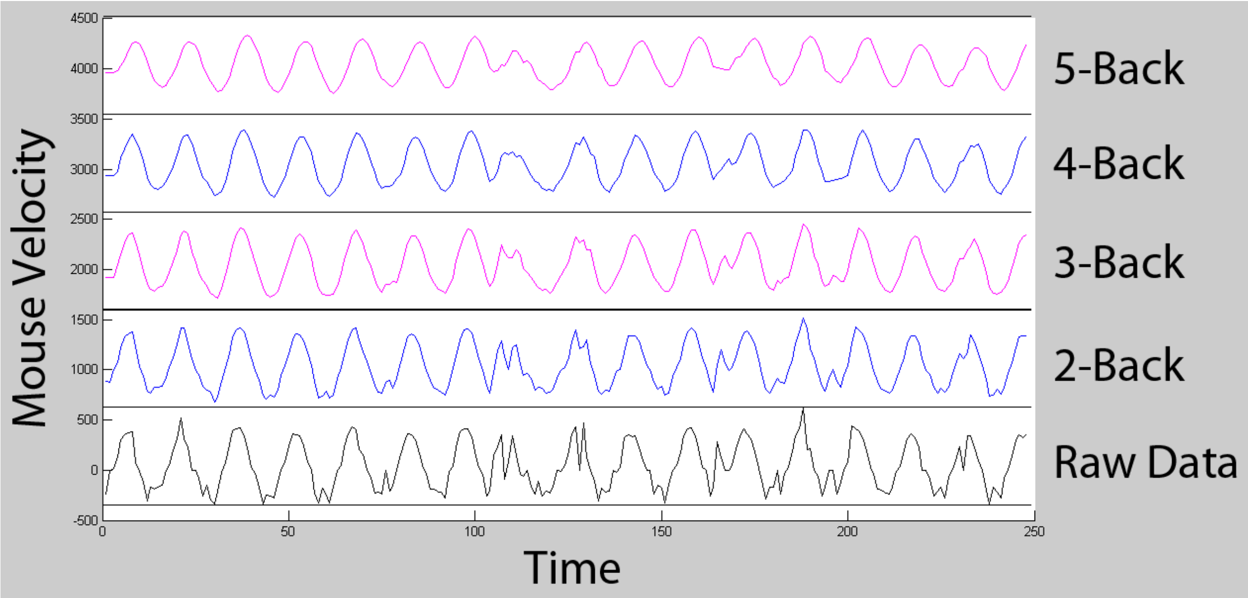
$$v_{(t)} = abs(\frac{p_{(t)} + p_{(t-1)}}{2})$$

Using this velocity variable as the input to the surface function requires that the acceleration is derived.

The velocity equation is an example of a 2-back interpolation and can be expanded to ***n***-back, where the velocity is averaged over ***n*** samples.

$$a_{(t)} = abs(\frac{1}{n} * \ v_{(t)} + \frac{1}{n} * \ v_{(t-1)} + \cdots + \frac{1}{n} * \ v_{(t-(n-1))} + \frac{1}{n} * \ v_{(t-n)})$$

A higher value for ***n*** gives a smoother velocity response but at the expense of responsiveness. Given that the sample period for the mouse position is (1000 ms / 60 Hz) 16ms, an n of 5 will cause a lag of 80 ms if the mouse is stopped suddenly (where mouse acceleration is expected to immediately drop to 0). The graph below shows how higher values for ***n*** smooth the velocity data for an eighth note stroke at 120 bpm. The data used here is a subset of the data from [Figure 10] and is 2 bars in length.



This data will be applied to the input of the haptic model as force = mass * $\boldsymbol{a}$. The mass is a user definable parameter and would, in the real world, describe the mass of the stroking implement and some aspects of the player's physiology. For the moment assume that mass = 1 and that force, $f_{lateral} = \boldsymbol{a}$.

As the stylus is pulled forward and back, the acceleration-deceleration curve can clearly be observed in the raw data and each curve is relatively uniform. However, as the stylus returns to a velocity of 0, as the stroking direction changes, some noise is evident in most of the strokes. Notice that in the raw data trace, there are several points where $\boldsymbol{a}$=0, this is not true for any of the interpolated traces. Each subsequent ***n*** has the effect of reducing the dynamic range of the data which would require scaling to fit to the 0 to 1 range of the surface function. The slight jitter at each direction change is relatively smoothed out by the 3-back filter and implementing this filter would incur a delay of ~50ms which may cause incongruent effects, particularly at the point of direction change

Implementing a low-pass filter in Pd is trivial and can also be used to smooth out the velocity response without incurring any additional delay. Several filters can be stacked in series in order to increase the order of the filter which will cause the slope to be sharper.

Interfacing

The simulations works in two development environments running simultaneously; PureData and Processing. PureData (Pd) is a visual environment for audio processing where patches are created by connecting operational blocks to one another. Processing is a Java-like language development environment with built in libraries for graphics, sound, networking etc. All of the simulation audio-haptic parameters are sampled, manipulated and output by Pd and Processing allows the GUI to be displayed.

The Processing program also manipulates mouse data at the Operating System level - this is necessary in order to prevent the mouse from reaching the edge of the screen where it will no longer send data. The Open GL “robot” class, for performing 3D transformations, allows the cursor to be set to any pixel value in the GUI window. The mouse position is reset every frame and the pixel difference used to derive the input lateral force and position.

Haptically Extended Mouse

The hand-held part of the device is a small travel-type computer USB mouse with integrated actuators and sensors. This connects to a breakout box with connections to and from the audio interface and power supply. The diagram below shows a schematic layout of the system.

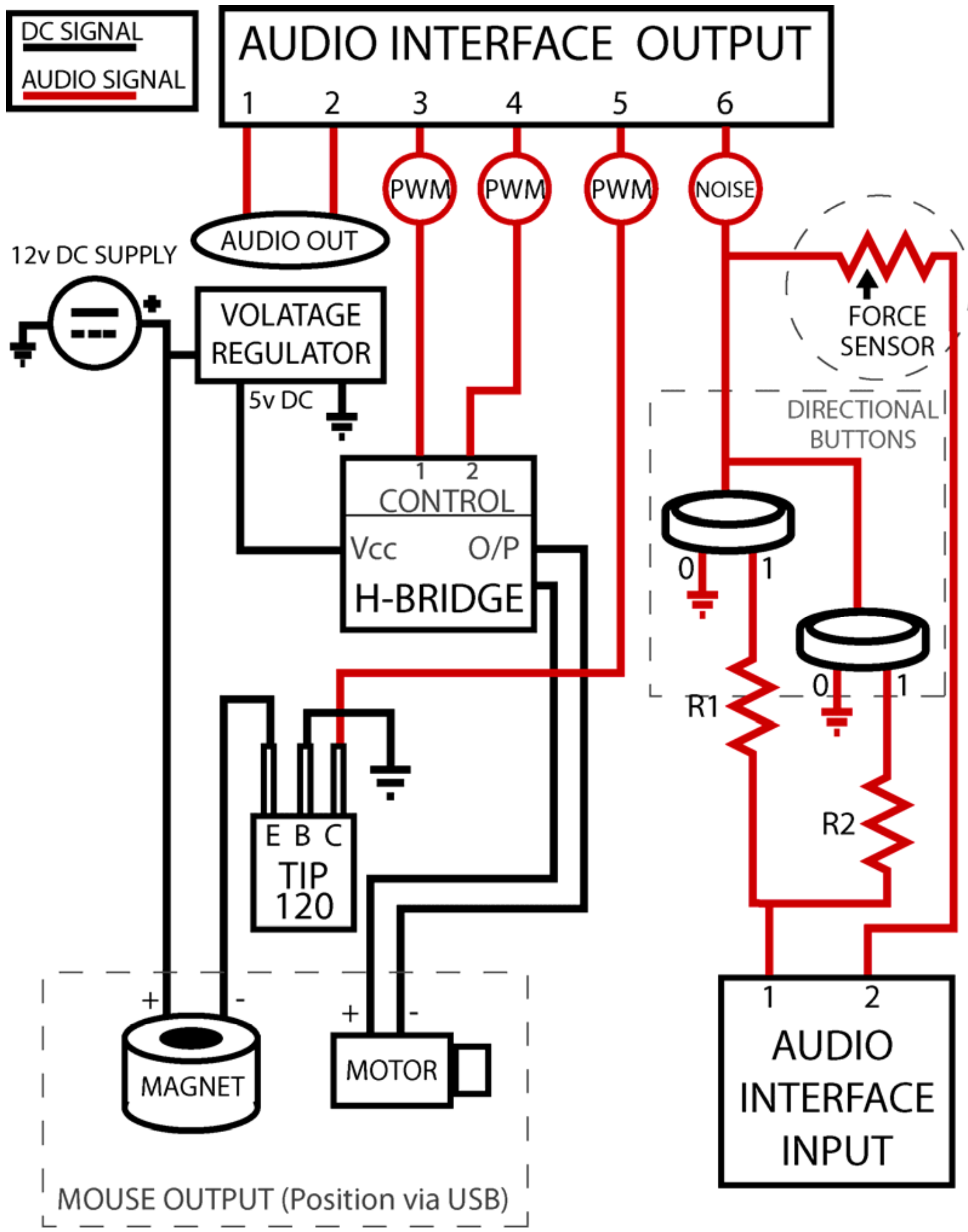


Figure 12 Schematic layout of the system

The haptic mouse consists of the mouse, two 25mm * 20mm electromagnets, one 20mm * 15 mm eccentric motor, one force sensitive resistor and two momentary push buttons. The magnets and mouse both have a very low tolerance for functioning above the surface and must therefore always be as tightly coupled to the surface as possible. Above 2mm, the magnets will not attract the surface and the mouse tracking does not work.

Informed by the prototyping session, a sensor and actuator system was designed that could fit into a base plate with a mouse. This allows removal of the mouse if one wishes to use it normally and also permits easier trouble shooting to the hardware if necessary. The unit is installed in front of the mouse and the index finger rests on the plate that houses the FSR.

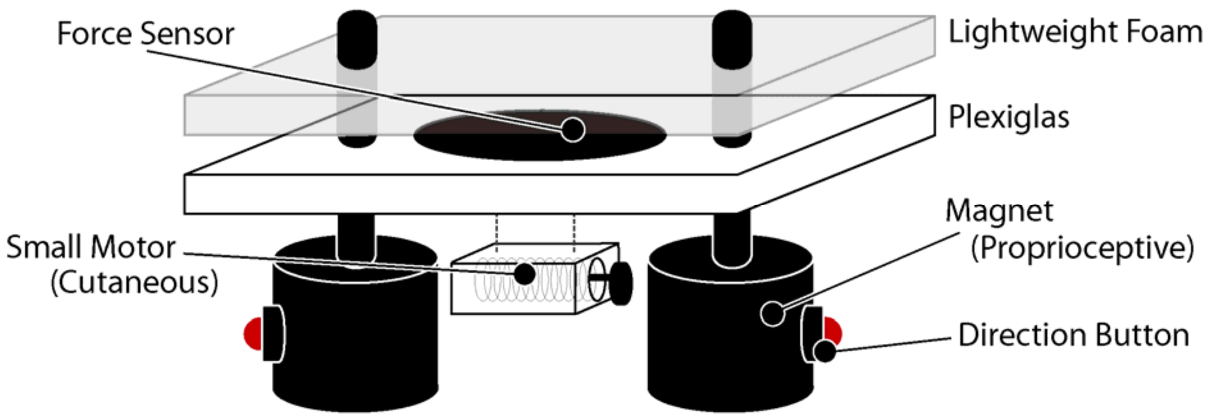


Figure 13 Sensor and actuator hardware design

Buttons

The buttons are implemented as a simple parallel voltage divider to transform an audio signal sent from the DAC that is then re-sampled. Each button has an audio signal from a DAC output on one terminal and a resistor to an output on the other terminal. The resistors, *R1* and *R2*, pass current to the output transformed by either independently or by both in parallel ((R1+R2)/(R1*R2)). Several audio signals were tested as potential sources; white noise, sinusoid and square wave. The noise signal, perhaps unsurprisingly, exhibited too much noise when re-sampled causing overlap in the ranges of each RMS value. The other two periodic signals both allowed for stable measurement and accurate detection of each button combination.

The table shows the values of the chosen resistors and the RMS of the transformed signal, where the original signal RMS is 0.70.

Resistor Value (Ohms)	R1 = 1000	R2 = 1500	R1|R2 = 600
Signal RMS	0.81	0.74	0.83

Table 1 Transformed signal RMS amplitudes per resistor value

The output is connected to a line level input on the audio interface and the RMS amplitude measured as a rolling-average. Determining which buttons are being pressed is done by checking against the stored value and is then put into a button state variable with 4 settings; None, Left, Right and Both.

Normal Force Sensor

The force sensor is similarly implemented. The sensor is a 6mm thin disc with protruding leads and is mounted on top of the mouse on a stiff plate between the magnets. This allows some pressure to be exerted without pushing the magnet down into the surface too much and creating extra friction. The same audio signal being sent to the buttons is passed to one leg of the force sensor and the other leg passes to another AD input. The audio signal is transformed by the force sensor and re-sampled into the system where the RMS amplitude is measured in the range 0.01 to 0.8 (signals are quantised by Pd to the range (-1,1)). This new value is mapped to the 0 to 1 Normal Force (f_N) range.

The sensor is most effective in the lower range of possible pressure, and reaches its maximum resistance with relatively little force - the pressure one would use to write firmly with a pencil is enough to push the sensor to maximum. It is unlikely that a user would exert a lot of normal force due to the discomfort that would entail (bending from the wrist while holding the side buttons) so the sensor is best suited to movements from the first knuckle. The position of the sensor plate accommodates the index finger and a pad of 5mm foam is attached across the top.

Design Issues

The main issue with this actuator design is acoustic noise; this could be reduced by coupling the iron mat to the tabletop via some absorbing mass. Any compression of the mass by normal force should be avoided so the mass must be both absorbent in relatively inflexible. The second issue is the placement of the normal force sensor; this should ideally be further back, along with the direction buttons so that the mouse may be gripped more tightly.

Actuator Control

Typically, actuator control is achieved by varying a voltage across an actuator, for example, the terminals of a motor or electromagnet. The speed, strength and torque of an actuator can be manipulated by raising or lowering the voltage which results in a varying vibration or other haptic effect. Many studies and commercial devices use microcontrollers to control these voltages and these facilitate the design of an integrated solution with OEM software, with standardised computer interfacing. For example; the Arduino microcontroller interface connects via USB, can be used wirelessly and has a large library of pre-made scripts and programs, and the Phantom Omni, also USB, allows the user access to its parameters through a development kit. The user may then create haptic models and effects by manipulating these parameters in their chosen development environment (MAX/MSP, PureData, Java, C++).

In this implementation, I employ audio signals to control the state of the actuators. This offers several advantages over standard Arduino-type microcontroller systems; OEM software is not required as the actuators will respond to standard, line-level, audio signals as produced by the majority of audio interfaces with an Digital to Analogue Converter (DAC), the rate at which the actuator can be switched is limited by the Nyquist frequency of the interfaces Sample Rate (SR) - typically 44.1kHz but up to 96kHz for affordable commercial examples (hardware capable of higher SRs are available but not commonly used by computer-based musicians due to cost, lower multi-track bandwidth and higher computational requirements for tracking and playback), the response time of the actuator is defined by the one-way latency of the host system - this can be as low as 1ms.

Response time and audio-haptic synchrony have been shown to have a noticeable effect on both performance and enjoyment of an interaction [Richard] and using the same hardware for both sets of outputs allows precise synchronisation. In my previous work examining stylus and surface-mounted haptics in a stroke-able percussion simulation , I observed asynchrony between microcontroller based haptics and audio from an independent DAC. This asynchrony is difficult to remedy as it is typically non-linear; the audio output from the DAC is usually rigidly continuous with no drop-outs but the microcontroller output may lag due to processor scheduling conflicts when the host is strained. The duration of the lag can vary and occasionally causes multiple aggregate actuations as the microcontroller “catches up”. For many audio-haptic applications, such asynchrony may not be noticeable, i.e. warnings or notifications in a Smartphone app or a simple guitar simulation app, but when the number of consecutive audio-haptic events is high, such as the rapid stroking of a percussive surface, congruent audio and haptic feedback is of high importance.

In digital audio, amplitude is measured from the ‘top-down’, from -148dbFs (decibels full-scale) up to 0dbFs, the maximum possible amplitude that can be rendered by the DAC. The resolution of the amplitude is determined by the Bit Depth and the majority of consumer interfaces offer 16 and 24-bit as standard, this means each sample can be one of 2^16 values for 16-bit audio (65,536 values) and one of 16,777,216 values for 24-bit. In practical terms this allows for very smooth transitions between events at a wide range of frequencies.

Magnet Control

A simple transistor-based circuit is used to modulate a DC voltage with an audio signal. The Field Effect Transistor (FET), model TIP 120, is an integrated circuit that has many signal control applications. The transistor can act as a high speed switch for large currents and can be controlled by small currents in an amplifier configuration. I implement an amplifier that is controlled by audio signals to provide precise, responsive control over actuators.

The FET has 3 terminals; Gate, Drain and Source and has a junction between source and drain (SD) and another between source and gate (SG). When a small current is flowing through the SG junction, the SD junction is opened and passes a larger current.

Here, the source is connected to ground so it would make sense that any potential above 0v at the SG will open the FET, but there are some caveats; the FET requires a minimum voltage to open and there is a diminishing return at higher SG voltages which causes the output to drop due to increased input impedance. Therefore the resultant waveform is not a perfect facsimile of the input- the speed of switching and the SG voltage influence the output.

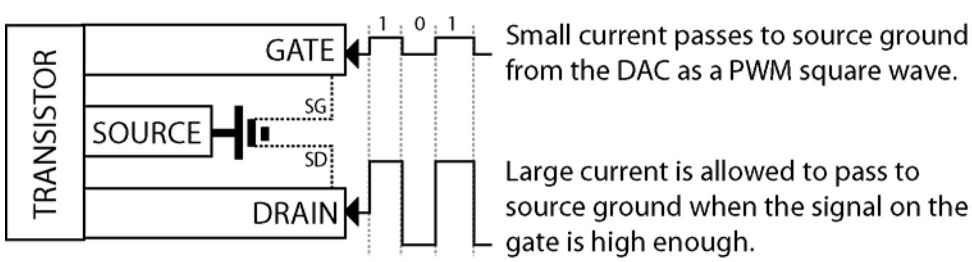


Figure 14 Modulating a high current with an audio signal using a FET.

A constant voltage is applied between one side of the actuator and the drain leg of the FET. As long as the current is not allowed to pass to ground the actuator will remain inactive. When an audio signal of 0dbFs (referenced to the source ground) is applied at the SG junction, the SD junction is opened and the current can flow through the actuator to ground, thus activating it. This arrangement is known as a source-follower and is analogous to an amplifier, where a low current is used to control a larger current as accurately as possible. By definition, an audio signal must be an AC waveform, so it is not possible to keep the transistor junction open at all times which would require a constant DC voltage. The result of this is that the maximum possible voltage applicable, and ergo the maximum haptic effect, is less than if using a microcontroller supplied DC source in the same circuit. In practice, the reduced range is adequate for most applications provided the actuator is over-specified actuators to offset this loss of level. That is, the maximum possible render-able voltage is in the upper range of the actuators requirements, i.e. one could use a higher-rated power supply to ensure the actuator gets enough power.

Motor Control

A small motor is used to provide cutaneous feedback to the users fingertip through the plate mounted above the magnets. The motor is controlled using an L293D integrated circuit which allows the direction of the motor to be changed using digital logic (HIGH and LOW, or +~1 vdc and 0 vdc). Aside from the power and ground connections, the chip has two inputs for logic control and two outputs for either motor terminal (+ and -). The table below shows how applying a high or low current to each input changes the behaviour of the motor.

Input 1 -> Motor +	Input 2 -> Motor -	Result
HIGH	HIGH	Motor is immobile
HIGH	LOW	Motor turns in clockwise direction
LOW	HIGH	Motor turns in anticlockwise direction
LOW	LOW	Motor is immobile

Table 2 Control of a motor with an L293D chip

This has many applications in robotics, for example a wheel can be made to move a vehicle forwards or backwards, but in haptics the main application is to create a sharper haptic sensation by using the control to "brake" the motor. These types of small motors often have rise- and fall-times that only allow "soft" haptic events and which also limits how rapidly consecutive events can be rendered.

Control Signals

Motor Control

The motor that simulates cutaneous feedback is of a lower resolution and poorer time-domain response than the magnet, due to the inertia of the iron shaft once moving and the energy required to get it moving. However, a range of haptic events are render-able with the motor; buzzing/vibrating, and small impact simulations. When rendering an impact, by turning the motor on then quickly off again, the perceived amplitude of the impact is a function of its duration. That is to say it is not possible to render a very short, very strong impact. As noted with the magnet, lower frequencies of Pulse Wave provide the strongest haptic sensation although the audible sound of the motor is considerably less, due its' smaller size and power consumption.

In Pd,

Given two input channels [Table 1], an impact is rendered as follows;

Apply a Pulse Wave of 50Hz to input 1 and set input 2 to "0" rendered as a signal.

After some delay, invert the inputs (and thus motor direction).

After another, shorter delay, switch both inputs to 0.

This has the effect of starting the motor, letting it run for the length of the delay, switching the direction of the motor and then setting the voltage across the motor to 0v and so stopping it. The time to brake the motor is constant and after some brief experimentation is set to 5ms - enough time for the motor to stop and not start turning in the opposite direction. The delay between the time the motor is turned on and the time the direction of current is changed is what determines the strength of the haptic event.

Values below 20ms do cause the motor to turn, but are too soft to be noticeable. Above around 55ms, the signal is perceived as a sustained "tone" instead of a staccato impact. The 20 - 55 ms range can render a very soft to very noticeable impact event, but it should be noted that values of above 60ms provide a much stronger haptic sensation that may be useable for effects other than surface impacts.

Magnet Control

Control Signals are the audio waveforms used to modulate the DC voltage provided by the power supply. To mimic the effect of a switched DC signal at the input, it makes sense to use an audio waveform with as high an average (Real Mean Square) amplitude as possible. Using PureData as a development environment, simple waveforms can easily and cheaply be rendered. Three waveforms are tested at varying frequencies to determine the output voltage. This manifests itself as the strength of the actuator, in this case how much grip the magnet has on the surface. The magnet requires around 4 volts DC to grip the surface enough as to be immovable for the user. The power supply is a standard 12 volts DC "wall-wart" type.

The graph below shows the resulting DC voltage at the output (source-drain) when the three waveforms are used as input at varying frequencies.

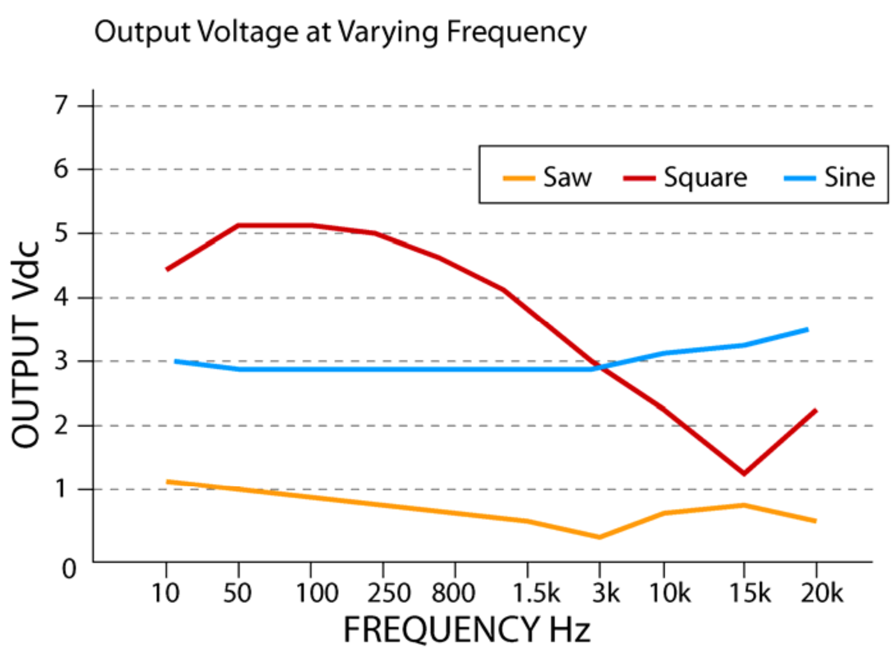


Figure 15 Output voltage with varying control signal frequency per waveform.

The square wave shows the overall best performance due to negating the minimum voltage required to open the FET, as it cycles from fully on to fully off. The sine and sawtooth wave are only opening the FET above the midpoint of their cycle so the amount of time the "on" signal is produced is considerably less.

Only the square wave signal can provide enough current for the magnet to be immovable. This feature is useful for rendering solid walls and very stiff surfaces.

Noise

The magnet makes some audible noise as it grips the surface in the form of a rattling-type vibration, the frequency of which is a function of the controlling wave frequency. This is clearly audible, but less so when using headphones and a full-scale constantly applied signal also represents the extremes of the system. Using a Shure SM57 dynamic microphone 5 cm from the magnet, I recorded the automatically generated vibrations from 9 different frequencies and measured the RMS value back into PureData using the *rms~* object on the AD input. It should be noted that the microphone and pre-amplifier response is not linear and will influence the frequency domain of the signal but this is constant for all signals. The results are consistent with what I expected from listening from the users point of view. The graph below shows the results of these measurements.

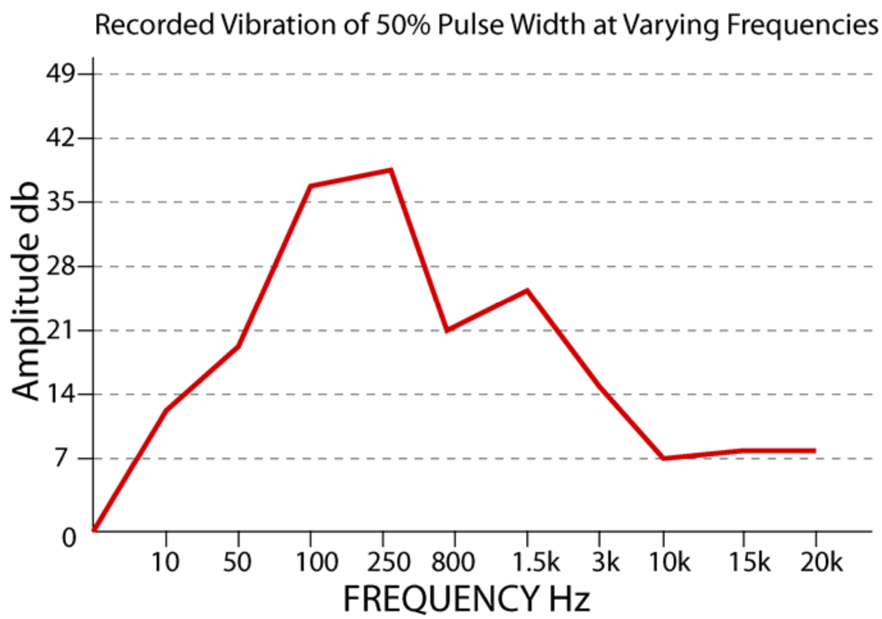


Figure 16 Recorded amplitude of acoustic vibration at vvarying fequency of control signal

The slight peak around 1.5 kHz may be due to increased resonance of the iron plate or may be a microphone resonance. A more accurate measurement could be performed in more controlled conditions but this data serves as a useful indicator of comparative noise across frequencies.

In the playing position, frequencies from 500 Hz and up are more distracting and become irritating when sustained. Applying the equal loudness contour to the data gives a better idea of how the vibrations sound to the ear [Figure 17] The contour shows that our ears are less sensitive to noise in the bass region, or more precisely; lower frequencies require more energy to be perceived as equally loud as higher frequencies.

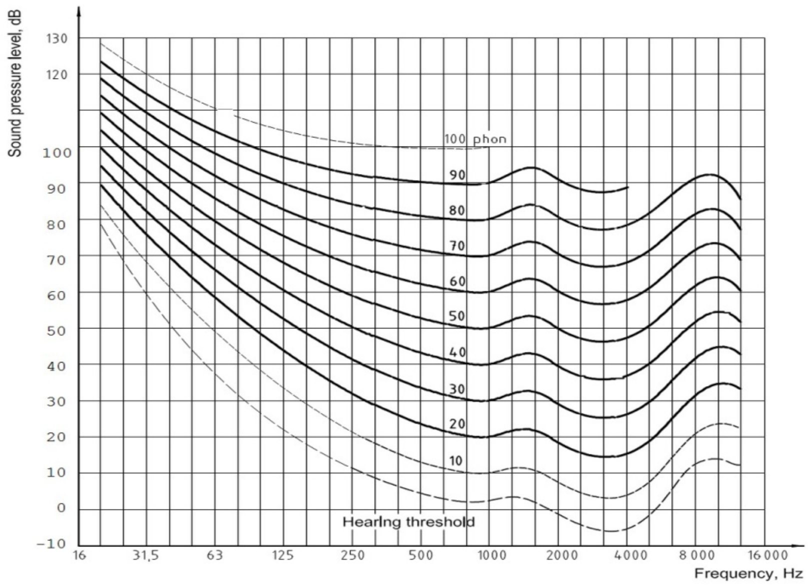


Figure 17 ISO 226:2003 Equal Loudness Contours.

The ear is particularly sensitive to midrange frequencies from approximately 500 Hz to 5 kHz, and this goes some way to explaining why the vibrations in this range are most distracting and leads to the conclusion that the control signal frequency should ideally be kept out of this range. To my ears, frequencies below 150Hz are the least unpleasant to hear and also offer the highest output voltage. Below 30Hz, I perceive the cycles as discrete events in both the audio and haptic domains.

It would be ideal to be able to move the signal above the range of human hearing but this presents some issues; the maximum strength of the magnet decreases significantly with increased control signal frequency and the possible resolution for Pulse Width Modification (PWM) is reduced as frequency increases.

Pulse Width

The Pulse Width of a square wave determines what percentage of its cycle is positive, a 50% Duty Cycle is a perfect square wave, on and off for an equal time per cycle. The RMS voltage increases with the Pulse Width and can be used to control the dynamic response of the actuator. Note that the signal RMS is equal to the PW/100.

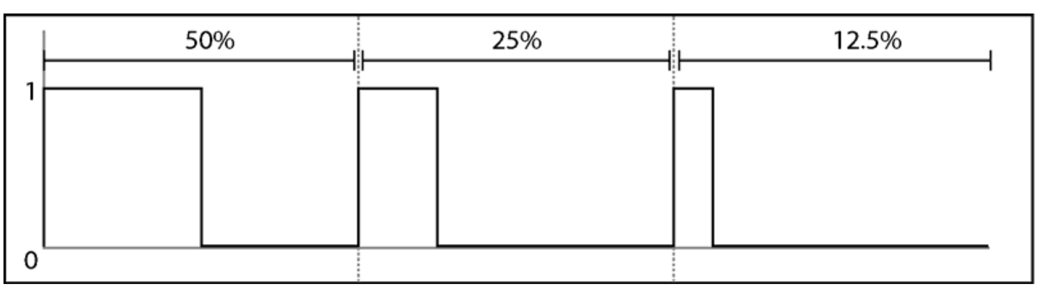


Figure 18 Pulse Waves with Pulse Widths of 50%, 25% and 12.5%

The graph below shows how PW influences the output signal at varying frequencies.

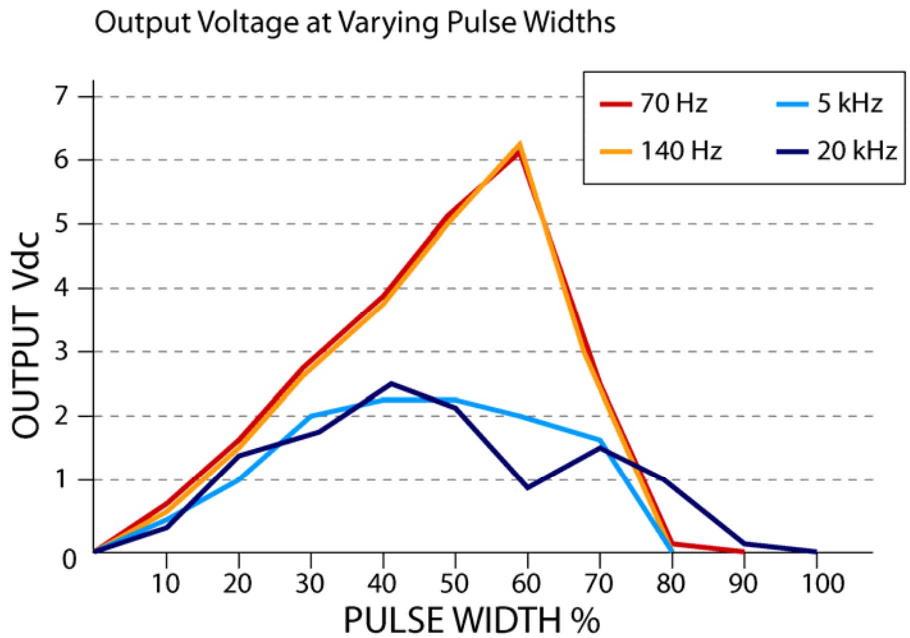


Figure 19 Output voltage at varying pulse widths

The DC decreases at higher input voltage RMS due to the FET input impedance becoming greater, above 70% PW, the output begins to fall sharply. The DC output follows the 70Hz and 140Hz signals Pulse Widths almost linearly up to 60%. Consider a perfectly linear relationship between the two parameters to be 10% PW to 1 volt DC. Up to 60% PW, the standard deviation from linear of the voltage of the 70Hz signal is 0.267vDC, and 0.226vDC for the 140Hz signal. Using such a low frequency as the control signal allows the haptic strength to be scaled accurately as the input voltage is consistent with perceived haptic resistance.

To get above the range of hearing, the control frequency must be above 20kHz. At the Nyquist frequency, half the sample rate (SR/2) and 22050 Hz in this case, the PWM can only ever be 50% as any wave is represented by only 2 consecutive samples. At the next possible integer division (SR/3) only two values are possible, 33% and 66%, and the frequency is 14.7 kHz and already in hearing range.

At frequencies of 15 kHz and up, the grip of the magnet is very slight so I conclude that it is not possible to put the vibrations out of hearing range without reducing the maximum haptic effect to an unacceptably low level. As the PW and voltage relationship is pseudo-linear up to a point of diminishing returns, I consider a wider range of PW values to be more important than acoustic noise considerations for this experiment. This is mainly due to the ease of mapping signal PW to magnet strength, and therefore haptic resistance.

Low level vibrations at lower frequencies are not noticeable when using headphones or monitoring at reasonable levels with speakers. A frequency of 70Hz is chosen as having some safe buffer allowing for a full-stop with considerable user force at a PW of 60%. The PW range of ~10% to 60% allows smooth control of the magnet from very slight resistance to full stop and can be switched rapidly between values.

Summary

Consider the Pulse Width signal as a constant for a moment; when it is applied to the magnet with a constant Pulse Width of ***x***, the state of the magnet is steady and provides some resistance, ***r***, to the lateral force of the user as they push the magnet along the surface. Even though the signal is allowing alternating current to flow to the magnet, the user perceives the ***r*** as constant. The ***r*** changes with the Pulse Width of a 70 Hz square wave almost linearly in the range 10% to 60% and this range is scaled to 0 to 1 and henceforth termed Magnet Signal (MS).

The Surface Function

In this section, the haptic model for a simple surface will be implemented and iterated upon and I will show how a periodic function can be manipulated cheaply to create a range of effects.

The output of the function is haptic resistance, notated as ***r***.

Imagine the surface function as a “black box” that takes the users’ force data as input and converts it into a haptic data that creates the impression of a fixed surface over which the virtual stylus can travel. The function is periodic and repeats over a subset of the available space; a densely populated surface, a sandpaper simulation for example, would have a short period and many repetitions whereas a washboard would have fewer repetitions and a longer period.

By modulating the Pulse Width of the Magnet Signal various haptic effects can be created. For example; if the modulation is mapped to the movement across the surface, the illusion of a distressed surface is created, if modulation is mapped to normal force, stickiness and viscosity can be rendered.

This modulation can be done with Pulse Width Modulation (PWM) using the MS as the carrier and another signal as the modulator. The other signal is the function that defines the modelled surface in a PWM configuration; the haptic output is equal to the PW of the Magnet Signal modulated by the surface function.

$$r = f_{surface}(user\ input) * PW_{MS}$$

The simplest example of this is to imagine a sinusoidal surface as a sin function of the users’ position for some repeating 0 to 2PI range. This creates the effect of a number of even bumps on the surface. This is similar to the way a Guiro is designed. The number of times the (0,2PI) range is repeated is dependent on the Surface Frequency (SF), derived from the width of each cycle peak in millimetres.

To establish a starting point for the function to translate to the real world, taking the data from [Figure 9], a 20cm sweep is completed in around 5000 pixels in screen-space, approximately 250 pixels per cm; therefore a ***density*** of 1 is defined as a peak width of 250 pixels.

[Figure 20] shows a graphical representation of the model with a peak width of 250 pixels, which is mapped to a range of 0 to 1.

$$range\ \{0,1\} = range\{\frac{0}{peak\ width}, \frac{peak\ width}{peak\ width}\}$$

Some operations must be performed on the sine function to map it to the range required for modulating the other signal. First, the sine is phase shifted by 1.5*pi to make it start at 0, when the position on the range is 0, i.e. at the onset of a haptic event. Then the amplitude is shifted to the range (0,1) by dividing it by 2 and adding 0.5. [Figure] shows the result of this transformation in comparison to the untransformed cycle.   The range parameter is multiplied by 2*pi to map it to the range of the cycle and the position of the stylus from 0 to 1 determines the output haptic strength.

$$\text{Transformed Sine} = \text{f(x)} = \frac{\sin\,(x - (\pi * 1.5))}{2} + \ 0.5$$

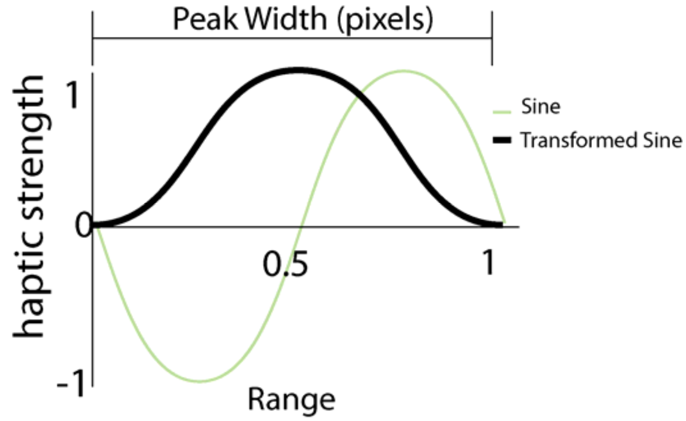


Figure 20 Transforming a sine wave for the Surface Function

The frequency of the cycle is determined by the stylus velocity; higher velocity will cause the peak to be rendered more times in a given sweep. This should be mapped as accurately to the 250 pixels per centimetre range as possible to maintain consistency with the user-defined density parameter. [Figure] shows that a medium speed sweep of 20cm is measured in 45 samples, the average velocity would therefore be;

$$\frac{20\ cm}{45\ * Sample\ Period\ secs} = \frac{20\ cm}{0.72\ s} = 27cm/s$$

If a surface with a peak width of 1cm is to be rendered, the frequency must be;

$$freq_{surf}Hz = \frac{velocity\ cm/s}{peak\ width\ cm}$$

And in this case the peak will be rendered at 27Hz for the duration of a perfectly even sweep. The peak width in pixels is determined by the density (ridges per centimetre) parameter which is user definable.

Multiplying the Magnet Signal with the current surface function will render an even surface with no dynamic effects, each stroke on the surface is rendered with the same haptic resistance and there is no impact simulation. Most importantly, the function will cause the haptic resistance to ramp down to 0 as the stylus travels down the slope of the bump. This is a somewhat unrealistic result, perhaps akin to a hydraulic lift; the effect of gravity is negated and movement in the downward direction is “padded” by the continually decreasing value of the haptic resistance. The haptic resistance variable is termed ***r***.

In a real interaction, the resistance would drop to zero after the peak of the bump was passed, and become negative to account for gravity. The sinusoidal function works as a graphic representation of the surface, and as a starting point for the haptic model but feels too soft and lacks dynamics.

Stylus Behaviour

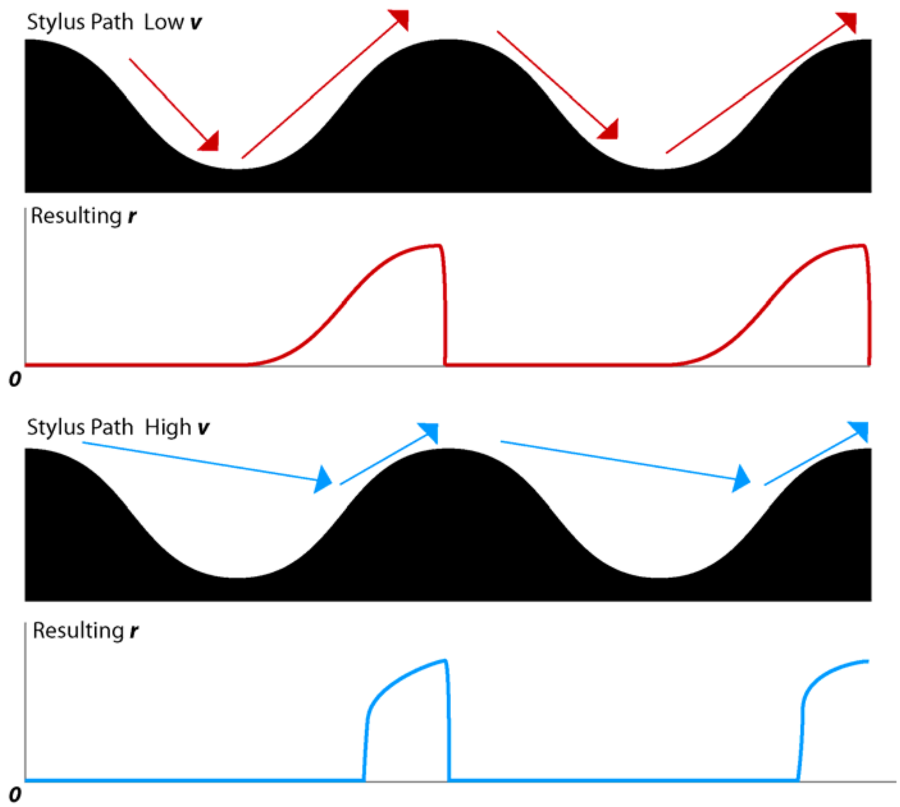


Figure 21 Graphic representation of how the path of the stylus changes with velocity and how this may be simulated with *r*

[Figure 10] shows how the stylus jumps from one peak to another at higher velocities and how ***r*** drops to 0 after the peak has been reached. For simplicity, gravity effects are not rendered (although this would be easy to implement by setting the base ***r*** to a value above 0). From the peak of a wave, haptic resistance should be 0 for a longer distance at high velocities and then quickly ramp to a higher value (blue trace). This is what produces the difference between the "rubbing" feeling at low velocities and the rattling/vibrating sensation at higher velocities; the discrete impacts at higher velocities are caused by the stylus jumping from one peak to the rising edge of the next.

If haptic strength is mapped to this new trajectory, the wave tends from sinusoidal to square. [Figure] shows how haptic strength (blue trace) could be rendered for a high v.

Notice also that a delay is incurred for the onset of the high v haptic event; the stylus must be some distance “up the slope” before it strikes it. The events are also shorter, resulting in a more discreet series of events rather than a constant rubbing. The lateral input force must therefore control the following parameters of the function;

- The number of harmonics (determines the square-ness of the waveform)
- The frequency of the function (determines the duration of the event)
- The delay before the wave is rendered (determines when the stylus hits the slope)

Manipulating the frequency of the function will also cause the density (bumps per centimetre) to change which is not desired, the spacing of the bumps should remain uniform. It would be possible to use delays and increase the number of instances of function to create this effect. Delays are often kept to a minimum in simulations as they can be expensive to compute. Several extra oscillators would have to be employed to get a sharp square wave; a perfect square wave is an infinite series of odd sine harmonics. This approach may therefore be unpractical for this simulation.

Looking at [Figure 21], a similarity to a Pulse Width Modified square wave can be observed. Using a PWM wave would eliminate the need to change frequency; the shorter duration occurs as a function of pulse width. However, it does not address the issue of the delay as the function is rendered at 1 when t is 0. This may cause the sensation of the surface sliding back as velocity is increased.

In terms of the function, the ***r*** must immediately fall to 0 after the peaking at 1. This will render the “half wave” shape shown in [Figure 21]. This is implemented by monitoring the direction of the function; the current and previous samples are compared until the function begins to descend, at which point ***r*** is set to 0 until the next iteration of the function.

In this implementation, I choose to ignore the delay that would be incurred as the stylus jumps at higher velocities. In practice, the distance jumped would be less than 3mm on a standard Guiro (1cm per peak) and , given the average velocity of 27 cm/s, the delay would be less than 8 ms and most likely unnoticeable. The pertinent factors here are the shape of the slope and the “hardness” of the haptic impact.

PureData Implementation

In Pd, a cheap solution is to blend the two signals together; the sinusoidal function and the pulse width equivalent. The pulse width is created by using a conditional operation on a phasor~ oscillator. At low velocities the blend is 1:0, sinusoid to PW, at higher velocities, the blend will tend to 0:1 and produce a shorter and harder haptic event.

Working with a constant ***v***, and therefore a constant surface frequency, the initial shape of the function is built. To get the half-wave form shown in [Figure] a sinusoidal wave is multiplied by a phase-locked pulse wave which has the effect of reducing part of the sine cycle to an amplitude of 0. Both the sinusoid and pulse wave have the same frequency. As the sinusoid tends from 1 to 0, the pulse wave is 1. As the sinusoid climbs back from 0 to1, the pulse wave is 0 and so the output is half a sine cycle from 0 to 1. The figure below shows the two source waveforms and the resultant multiplication of the sine with the pulse waves (multiplied trace enlarged for clarity). This multiplication of signals is called Amplitude Modulation (AM) synthesis, and the sinusoid is termed the carrier, and the pulse wave the modulator.

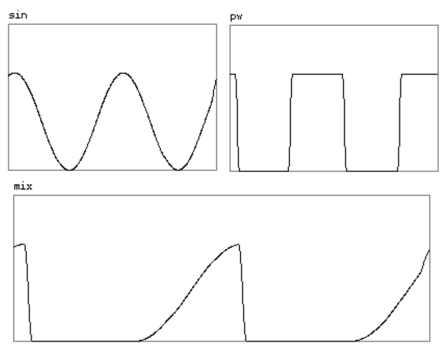


Figure 22 Resultant multiplication of two simple waveforms to simulate the haptic response of a sinusoidal surface

In order to create the sensation of jumping from peak to peak [Figure], the duration and square-ness of the wave must be changed as a function of velocity. The figure below shows the multiplied wave when the width of the pulse wave is 50% (top) and 10% (bottom).

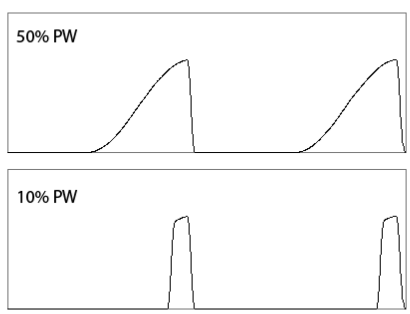


Figure 23 The influence of pulse width on the duration and envelope of the haptic event

From [Figure], there is a clear resemblance between the ideal trajectories and the function. The velocity is mapped to the pulse width as follows, which puts the PW range from 0.6 to 0.1;

$$HapticSignal_{PW} = abs\left(0.6 - \frac{v}{2}\right)$$

The overall amplitude is also mapped to ***v***, although most of the dynamics in the simulation must come from normal force. I allow ***v*** to vary the amplitude by 50% by mapping the 0 to 1 range to 0.5 to 1 and multiplying the output by this number. The trace below shows a slow sweep of ***v*** from to zero to, arbitrarily, 8. The bottom trace shows how ***v*** can be used in the function to simulate the dynamics the model requires.

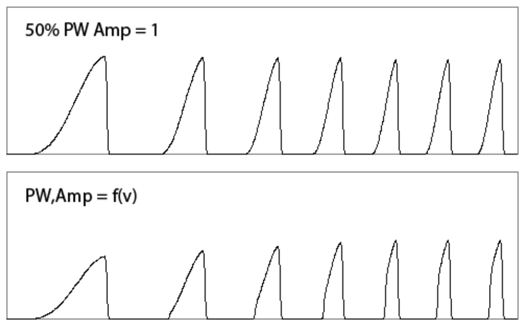


Figure 24 Mapping of *v* to PW and amplitude of the function

As long as each indentation on the surface is symmetrical, stroking direction is unimportant as, haptically speaking, the sequence of events is the same at every stylus-indentation interaction. The haptic resistance always increases from 0 to 1 at the onset of an event.

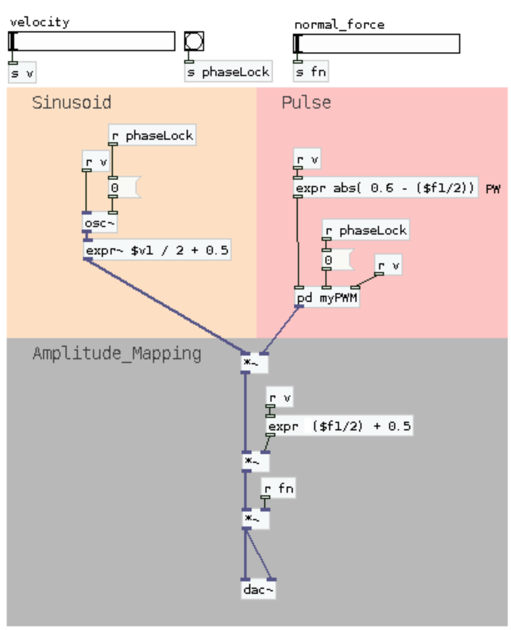


Figure 25 Pd implementation of the surface function

Both the topmost oscillators (osc~ and myPWM) take ***v*** as their frequency input, this means that creating a denser surface is simply a case of multiplying ***v*** with the Surface Density parameter. When the velocity drops to 0, the waveforms are phase-locked by having their phase reset to 0 degrees simultaneously. Phase-locking in this way during a stoke, i.e. at ***v*** of over 0, can cause glitches as the waves reset to their initial amplitudes so it is preferable to lock the phases at times where the mouse is not moving, for example on a change of direction.

The osc~ operator has an additional transformation to the 0-1 range that the PW does not require. The pulse width is set by the expr~ object which contains the equation for $HapticSignal_{PW}$. These two signals are multiplied together and then multiplied by ***v*** which is mapped to the 0.5 to 1 range. The above implementation also maps normal force linearly from 0 to 1 to amplitude which allows the overall dynamic to be controlled by finger pressure.

In order to change the shape of the surface, the carrier waveform may be changed in several ways to alter the feeling of the surface. If the frequency of the carrier is higher than the modulator, the shape of the function will change to accommodate more than half of the cycle. Practically, this results in an increase in surface density but the distance between events, or sets of events, is still the same with matched frequencies.

Modulating the carrier wave with another, higher frequency wave is how the roughness parameter is implemented. [Figure] shows the function of a ***v*** sweep from 0 to 5 with another waveform modulating the amplitude of the function by 30% (± 15%).

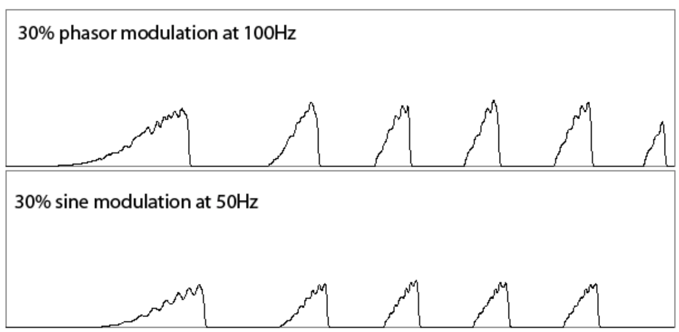


Figure 26 Modulating the function with a higher frequency waveform to simulate roughness

The ***r*** function for the topmost trace in [Figure 18] would be;

$$r = \left((pulse(v) * \sin(v)) * \frac{\sin(50)}{6} \right) * \left(\frac{F_L}{2} + \frac{1}{2}\right)$$

In this simulation, the following parameters are user definable;

- Surface Frequency - how many dips there are per centimetre
- Surface Depth - the maximum peak height of the surface
- Surface Roughness - The smoothness of the surface
- Body Size - The square size of the resonating body
- Body Materiel - What the resonating body is made of (wood, metal, plastic).

- Instrument Type - What type of instrument the simulation is of (Comb, Guiro, Washboard, Particle Drum)

The user inputs direction, lateral and normal force, and a button combination and the state of these variables may be displayed for debugging and monitoring purposes although these would be removed for the ultimate end user. Additional "housekeeping" parameters may also be adjusted; audio and haptic output levels, and force parameter scaling to more accurately map the lateral force sensor to the user; I expect men to generally be able to push harder than women and likewise adults and children.

As ***v*** is determining the surface function frequency at any given time, it follows that a transformation of ***v*** will alter the perceived resolution of the surface. If ***v*** is multiplied by a coefficient of greater than one, the resolution of the surface will increase, and decrease when the coefficient is less than one. Therefore the Surface Frequency parameter can be implemented as

$$v = v * surface_{freq}$$

Surface Depth is implemented by simply multiplying the entire surface function with the Surface Depth parameter. Roughness is implemented by modulating the amplitude of the function with a higher frequency waveform, the amplitude of which is determined by this parameter.

Body Size and Material Type are implemented by altering the delay line in the resonator and by changing the balance of the impact harmonics.

Instrument Type selects a preset based on the other parameters.

Rendering Stiff Walls

To implement solid walls, it is necessary to be able to derive direction independently of velocity [Salcudean]. That is to say one must be able to measure the *intended* direction while the solid wall is being rendered without having to allow some small movement to compare the current position to the last position. When the virtual stylus is at the wall boundary, movement in one direction should be impossible; this is easily rendered by applying full power to the magnet. However, if the movement in the opposite direction is measured, the magnet must be released and the stylus allowed to travel. The problem here is of course that the stylus cannot travel as long as it is at the boundary.

One possible solution could be to allow small movement by pulsing the magnet between the maximum value and a lower value and then setting the value to high if positive movement is measured and to low if negative movement is measured. The main disadvantage of this approach is that the wall will never be fully stiff. The stylus will be pushed into the wall each time the direction is measured, albeit by a small amount, a function of the pulse frequency.

It is then necessary to have an additional measurement of direction; positive or negative.

Hardware Implementation

GUI Design

The purpose of the GUI is to allow the user to make changes to the parameters of the simulation with the mouse and to provide visual feedback about the current simulation. The design is informed by currently available plug-in instruments for Digital Audio Workstations that share a similar feature set.

Figure 27 Typical virtual instrument GUIs. (Left to Right) Modartt Pianoteq, ASS Lounge Lizard and AAS Tassman

Each of these examples is a physically-modelled instrument plug-in that recreates the parameters of a musical instrument. Regarding the GUIs, each example displays an image of the current model and a sub set of editable parameters (buttons, sliders, menus). Each design has two main components; state of current simulation to some degree and a set of editable elements.

The state of the current simulation is also displayed, as a moving puck along a repeated waveform. The GUI surface parameters affect the rendering of the waveform and work as a stylised version of the current Surface Function (it would impractical to exactly draw the SF). The puck moves with the mouse, and is redrawn to loop around at either extreme, allowing the user to see the workings of the simulation as they use it. The waveform also allows users to see in real time the influence of adjusting the surface parameters.For this work, I am less concerned with screen resolution than legibility to I choose to make the GUI large in comparison to a typical instrument plug-in. The most important consideration is that the user can navigate the GUI efficiently and that it offers clear edit-ability.

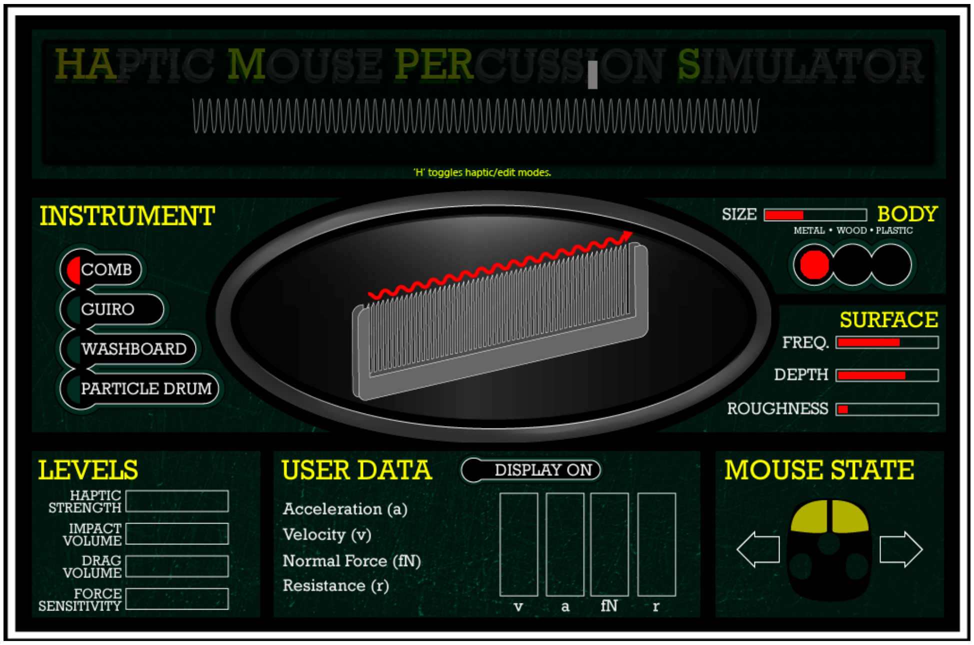


Figure 28 GUI for the simulation, showing current state and editable parameters.

A more accurate representation of the GUI as it would be when conforming to the standard of typical VST or AU instrument plug-ins is shown below.

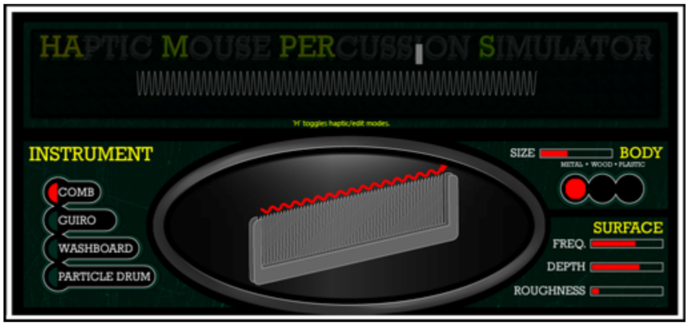


Figure 29 Mockup of the GUI for integrated DAW use.

Interaction examples

Guiro

Description

The Guiro family of instruments are made of a closed cylinder with ridges across the short axis on either side. Guiros have traditionally been made from the dried husks of fruits and vegetables, typically gourds, and are now available mass-produced in plastic and metal. The range of interaction is limited by the length of the Guiro's playing surface, normally around 25cm by 7cm for a modern steel model.

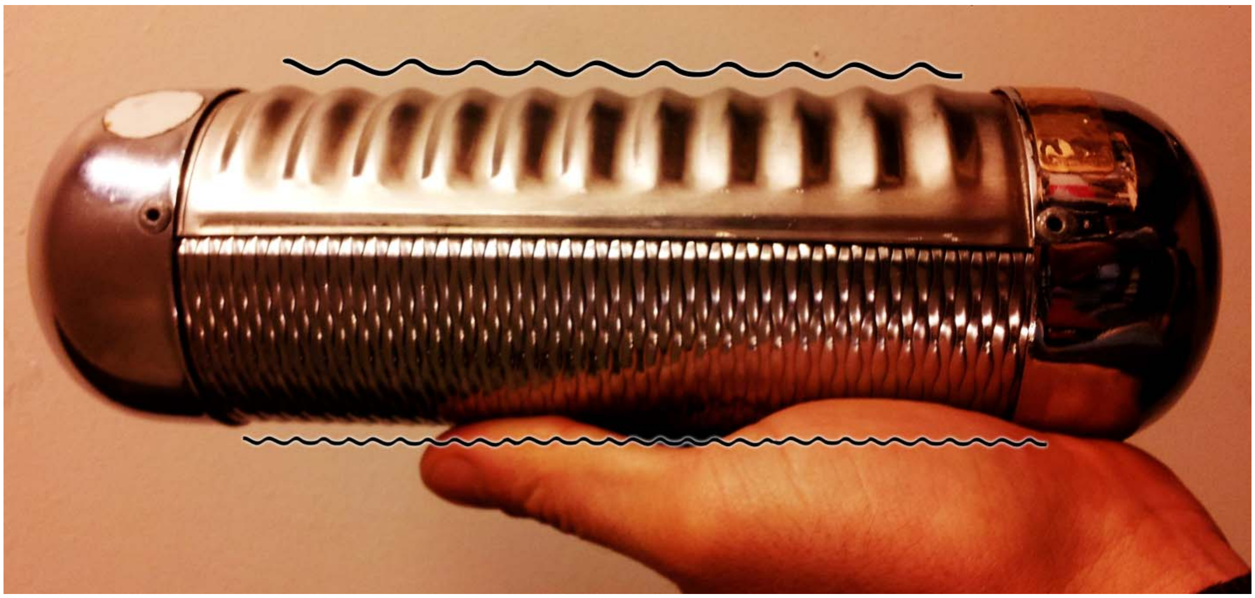


Figure 30 Typical modern Guiro with two sinusoidal deformed sides

Playing Style

The Guiro is played with either a drumstick or with a stiff brush and can be struck or rubbed to produce sound. Skilled players oscillate their movements so that the impacts are locked into a tempo, and usually at a subdivision of at least 16th notes. The player can then control the dynamic of their playing by altering the pressure between the stylus and surface. Additionally, the player can shake the Guiro, agitating the beans and create a semi-continuous swishing effect or a discrete, but decaying, rattle.

The amount of energy required to shake the Guiro enough to audibly agitate the beans is low, one only needs to give a slight jerk of the wrist, moving the Guiro perhaps 2 or 3 cm around its' own axis. A harder shake produces a louder initial impact and a longer decay as the beans settle to the bottom.

It should be noted that continuous hard shaking while playing the Guiro with the stylus is not normal playing practice; shaking the Guiro hard at low subdivisions, below 8th notes, is usually done without any stick interaction, in the style of maracas. Playing with the stylus normally results in a rhythmically busier part and offers great scope for accents and dynamics.

The Haptic Dimension

The Guiro interaction can be broken down into three modes of proprioceptive interaction;

- Striking an (for these purposes) infinitely hard surface.
- Providing an inertial “shove” to constrained bodies.
- Traversing a periodically-distressed surface.

It can also be said that lateral movement of the arm without stylus/surface interaction is a form of proprioceptive interaction but it does not concern the model at hand. It is essentially “modelled by not being modelled”.

The cutaneous feedback comes from the stylus impact and is coincident with the proprioceptive feedback in all instances. This may reduce the need for a very accurate cutaneous model.

Sound

The sound of the Guiro is made up of impacts with resonance from the body and the slight movement of the beans at each impact. That is; a short high pitched metallic impact that does not vary greatly with playing velocity but may be softened by reducing downward pressure and a short resonant sound from the body. The decay of the resonance is short due to the small body. The resonance may be damped by increasing one’s grip on the underside of the Guiro and this can also add to the sense of dynamics in a performance.

Mapping

The model requires three types of input; strike, shake and rub. Rubbing is done by simply mapping the mouse velocity to the surface function.

Shaking is a threefold process; first the surface must be grabbed, then moved and then finally released. The grabbing action is done by simultaneously pushing the buttons on the sides of the mouse in a pinching motion. This motion is very similar to gripping the physical Guiro, although the position of the hand is inverted. When the mouse is grabbed, the rubbing model is made inactive and the system then waits for one of the buttons to be released, at which point the last velocity is passed as the initial force input to the beans model.

Striking is mapped to the normal force sensor and is does by tapping it quickly, although the low sample rate prevents accurate velocity sensing for such a short event. Finally, stylus pressure is mapped to normal force.

One potentially incongruous aspect of this mapping is that the instrument is now played entirely one-handed which may lead to fatigue and also removes the independence between the shaking and striking actions.

Audio

As a starting point, a Guiro was played and recorded at various dynamic levels. Analysing these data can help to give an idea of the makeup of the sound and how it varies with velocity. Form a listening standpoint, the sound is best characterised as a high pitched click with body resonance. The effect of increased velocity on the sound is to make it “harder” which can be quantified as; more high frequency content, higher amplitude and a longer decay.

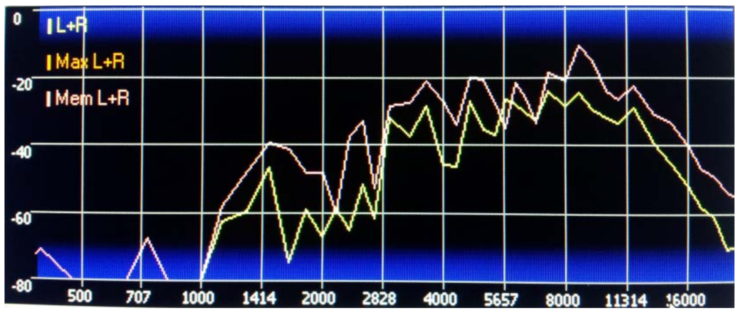


Figure 31 The frequency response of two isolated Guiro impacts during a stroke.

A series of stokes were recorded and a typical example used as the basis of an analysis. [Figure 14] shows the frequency response of a typically soft (yellow trace) and typically hard impact isolated from a longer stroke. For all parts of the signal above -60dbFS, peaks can be observed at approximately; 1450Hz, 2400Hz, 2900Hz, 3600Hz, 4500Hz, 5800Hz, 7250Hz, 8600Hz and 12000Hz. This shows that the sound is pseudo-pitched; there are several peaks that are close to being integer multiples of one another. The annotated trace below shows a close harmonic relationship between the even-order modes of 1450Hz (yellow lines). An additional two harmonic set is found at 3600Hz and 7200Hz and two further peaks do not fit into these series, 2500Hz and 4800Hz.

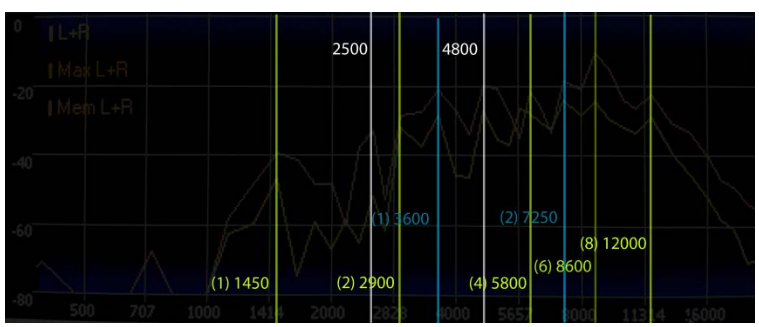


Figure 32 Annotated frequency response showing the harmonic content of the Guiro strike

Because the harmonic partials are not exact integer multiples, the resulting sound is metallic and of an indiscriminate pitch. If the intervals were closer to being multiples, the tone would tend to being more bell-like.

The envelope of the sound is very short, approximately 6 milliseconds after the initial strike the sound is inaudible. The envelope of the harder strike is a little longer, the body of the sound lasts 9ms.

The impacts are modelled using additive synthesis and a resonator. The overall amplitude of the impacts is determined by normal force and, to a lesser degree, lateral force as discussed in [].

A series of sine oscillators are employed at the frequencies derived from [Figure 15]. Two noise generators, band-passed at 1500Hz and 6000Hz are used to fill in the missing information from the sinusoidal signals.

A simple resonating delay line emulates the body resonance and uses the lowest four harmonics as an impulse. The impulse is fed into a delay, which is set to a very short time, and then the output of the delay is attenuated, filtered and fed back into the input. This causes the impulse to lose energy over time and decay as a function of the filter profile and damping coefficient. For this model, the impulse is filtered at 1500Hz and the damping is 0.99. The delay time is set to 4ms which produces a lower tone than any of the additive oscillators. The effect and purpose of this is to add body to the sound, and not to make a noticeably resonant body, so a simple solution is both adequate and resource friendly.

These signals are combined through a linear Attack-Decay (AD) envelope to the audio output.

Normal force is mapped linearly to the output amplitude from 0 to 1 which is also affected by lateral velocity;

$$audio_{amp} = F_N * (0.8 + (F_L * 0.2))$$

Additionally, the signal is high-passed filtered at a frequency of;

$$highpass_{freq}\ Hz = 1000 - (F_N * 1000)$$

Finally, the length of the decay in the AD envelope is determined by normal force, which causes the decay to vary between 6 and 9 milliseconds;

$$env_{decay}\ ms = 6 + (F_N * 3)$$

This is a cheap way to emulate the subtle interaction between the impact velocity and resonance of the body, as harder strikes result in more resonance. Lateral force is not included in this term as it, counter-intuitively perhaps, has a much lesser influence on the sound than normal force. Using normal force as the main parameter for amplitude also makes the simulation easier to control.

Washboard

Figure 33 Small Washboard, suitable for percussion

The washboard is a reduced amplitude version of the guiro but with some important changes. The function is reduced in dynamic range and decreased in frequency. The harmonics in the audio simulation are more closely aligned to integer multiples which produces a purer sound. The frequency of the resonance is lowered by reducing the delay time in the resonator; the signal is also low passed at 1800Hz to produce a lower, thicker sound than the Guiro.

Although the playing styles are similar, the mode of interaction is different than the Guiro. The washboard is either worn around the next so the playing surface is parallel with the chest or it is straddled in a seating position.

Solid Walls

Solid walls can be rendered by applying full power until a break condition is met. In this case the break condition is that the button on the same side as the wall is pressed. The interaction is designed as having some space to accelerate before hitting the wall. On the detection of a single direction button, the system derives position from $\boldsymbol{v}$ and applies $\boldsymbol{r} = 1$ after some distance has passed. When the opposite button is pressed alone, $\boldsymbol{r}$ is reset to 0 and the mouse allowed to travel.

A table-type audio model is used, based on the Karplus-Strong algorithm [Karplus]. The Karplus-Strong (KS) algorithm uses a recirculating delay to mimic the decay characteristics of a resonating object. A short initial impulse, either a single high-frequency cycle or noise burst, is used as the initial force impulse in the model. The impulse is then delayed by a time in samples, multiplied by a dampening coefficient and low-pass filtered, and then both sent to the output and fed back into the delay line again. To extend the frequency resolution of the algorithm, it must be possible for the delay time to be a non-integer multiple of the sampling rate. This is made possible by interpolating between the floating-point time intervals so that they may be approximated in the discrete-time domain.

If the delay time is constant through the decay stage, the resulting sound will be pitched and resemble a string or bell. Increasing the delay time during the recirculating phase causes the signal to decay much faster and resemble a drum or transient strike.

Particle Drum

Description

In addition to allowing simulations of existing instruments, the system can be used to create new instruments that could not be realised in the real world. The Particle Drum is a modelled percussion instrument that defies the laws of physics and has more in common with a string instrument.

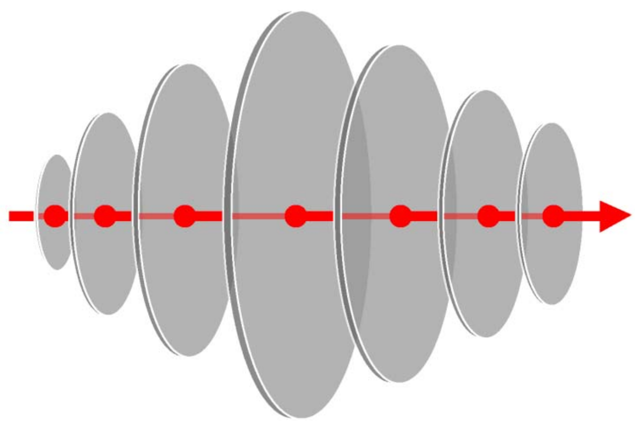


Figure 34 Graphical representation of the Particle Drum

The instrument consists of several tensioned drum heads fixed along an imaginary axis in space. When the user passed through a drum head, the effect is a string-like resistance-then-break action. The tension of a drum head determines its pitch, haptic resistance and the force required to break through it.

Playing Styles

As the instrument has no physical analogue, it is perhaps not apparent what range of techniques could be applied to playing it. Playing styles applicable to other instruments may serve as a basis for interaction, for example a glockenspiel type sweep could be modelled by mapping normal force to the Z-axis of the stylus. The user can oscillate between a subset of the heads to produce rapid rolls and rattles or could sweep across them in much the same way as the Guiro interaction.

The Haptic Dimension

The behaviour of the particle drum is similar to a string; an initial force impulse is resisted, released and then causes vibration. The Mass-Spring-Dampener model can effectively model this.

The Mass-Spring-Dampener (MSD) model is an example of a Backward Finite Difference Model; the current *x(t)* position of the mass is a function of the system parameters and *x(t – delta t), and x(t –*

2delta t). The model simulates the behaviour of a mass attached to a dampened spring, fixed at one end. When force is applied to the mass, stretching the spring from its resting position an equal negative force is applied to the mass, causing it to move in the opposite direction, *-x*. The x position is changed over time as a function of t^2, which gives an exponential decay. Adding a dampening coefficient to the equation changes the decay time of the string. The frequency of the string is given by pi/2 * √(spring coefficient/mass) which are both usually constants for a given string.

Sounds

The Karplus-Strong table model is used with a different pitch assigned to consecutive drums.

Conclusion

The system is able to mimic the proprioceptive events of a stroke-able percussion interaction bur needs some additional refinement of the physical prototype before it can be field tested.

Some difficulty arises when attempting to play a combination of a low subdivision shaken sequence at the same time as a busier rubbed sequence. The need to pinch the device can cause a “hiccup” in the rhythm as the grip on the mouse is changed.

It is possible to derive short normal-direction impact velocities as a function of time but the sample rate is too low to allow striking at anything other than a binary resolution (one or off, striking or not-striking). This means the striking model must be binary. This works well when the tempo is fast and the rhythm is busy, but not as well at slower tempi or sparser rhythms. The lack of dynamics makes this an incongruous experience between the proprioceptive and audio domains, particularly when rubbing with a low normal force and then striking.

Mapping the cutaneous motor feedback to a strike is ineffective due to an easily perceivable delay between the physical strike and modelled haptic response. This is also a somewhat unnecessary mapping; the natural cutaneous feedback provides an adequate striking sensation.

As the system can also produce low-level vibrations, it is feasible that instruments that produce a continuous, time-varying vibration could be modelled. A violin string could be “bowed” back and forth with the mouse with a MIDI keyboard simulating the left hand and determining pitch. The varying stickiness of the string can be simulated by low PW values and could also be a function of normal force. In the same way rubbed bowls could be modelled using a stick-and-slip model and the simulation could be expanded into 2 dimensions.

The current system could also be expanded into 2 dimensions by simply applying the surface function to the mouse y velocity, thus rendering a symmetrical surface subdivided by square functions. A more practical use of the y range could be to set up zones with different surface parameters, i.e. both sides of the Guiro could be modelled with the finer surface “on top”. The current problems with velocity influencing the position variable would have to be dealt with in order to stabilise the zone positions.

Implementation of this system in a native Digital Audio Workstation environment is still some way off. The requirement to override the OS mouse control and to interface with the DAW in general presents several issues;

1. Button and force sensor data must be independently recorded as audio.
2. There must be two mono audio outputs, one for sound and one for haptics.
3. Mouse data must be able to be recorded into the DAW, probably in a native format i.e. audio or MIDI.
4. Any software simulation must therefore be able to parse the native data as well as the “live” mouse data. Or must be able to convert from live to native with minimal latency.
5. Ideally, the mouse data and some MIDI data from another source should be able to be recorded concurrently and share the same, if any, latency.

In my opinion, point four in the above list represents the greatest challenge; the continuous mouse data is what allows the simulation to be successful. Converting this data to MIDI Continuous Controller (CC) data may be possible but it is considerably lower in resolution than audio, which is also an option. This may be a problem due to the limitations of MIDI CC resolution, which is is 8-bit with audio-rate sampling. Some work would need to be done into the playback of mouse data as CC data to inform any changes in the audio-haptic model. It is likely that the interpolation algorithm would have to be changed to handle the lower resolution.

Using MIDI data would be ideal, as it would allow the model plug-in to be standardised to a native format (VST, AU) with one MIDI input (the parsed mouse data), two audio inputs (buttons and force) and two mono outputs.

It may be possible to send both the force and button data to one audio track, by modulating between them and detecting the different ranges (i.e. buttons always above 0.75, anything else is force), but a more elegant solution would be to send them in the USB stream and so eliminate the need for an audio input. The effect of the reduced sampling rate would certainly affect the data and further study could investigate if this is a possibility, as a system with a single USB input would be more user friendly and require less calibration.

References

[Adelstein] Bernard D. Adelstein, Durand R. Begault, Mark R. Anderson, Elizabeth M. Wenzel. 2003. “Sensitivity to Haptic-Audio Asynchrony” NASA Ames Research Center.

[Baille 2011]Lynne Baillie, David Beattie and Lee Morton “Feel What You Hear: Haptic Feedback as an Accompaniment to Mobile Music Playback” IWS ’11, August 30, 2011, Stockholm, Sweden.

[Bau 2010]Olivier Bau, Ivan Poupyrev, Ali Israr, Chris Harrison. “TeslaTouch: Electrovibration for Touch Surfaces” UIST’10, October 3–6, 2010, New York, New York, USA.

[Berdahl] Dan Overholt, Edgar Berdahl and Robert Hamilton. “Advancements in Actuated Musical Instruments.” 2011 Organised Sound 16(2): 154–165

[Botvinick] 1998Matthew Botvinick and Jonathan Cohen. “Rubber hands ‘feel’ touch that eyes see.” 1998. Nature, Vol. 391, 19 February 1998.

[Bongers 1998] A.J. Bongers. “Tactual display of sound properties in electronic musical instruments” 1998. Displays 18 (1998) 129-133.

[Cadoz] Claude Cadoz, Leszek Lisowski, Jean-Loup Florens. 1990. “A Modular Keyboard Design” Music Journal, Vol. 14, No. 2, New Performance Interfaces 2 (Summer,1990), pp. 47-51

[Chen] Chen, Jun, Chris DiMattia, Mike Falvo, Pichet Thiansathaporn, Richard Superfine and Russell M. Taylor II, 1997 "Sticking to the Point: A Friction and Adhesion Model for Simulated Surfaces." Proceedings of the Sixth Annual Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems 1997.

[Cook] Perry Cook. “Principles for Designing Computer Music Controllers” Department of Computer Science. Princeton University

[Gescheider] , G.A. Gescheider “Auditory and cutaneous temporal resolution of successive brief stimuli.” J. Exp. Psych. 75(4), 1967, 570-572.

[Hayward]Vincent Hayward. “A Brief Taxonomy of Tactile Illusions and Demonstrations That Can Be Done In a Hardware Store” Brain Research Bulletin, 2008, Vol 75, No 6, pp 742-752.

*[Hayward] Vincent Hayward and Karon Maclean. “*Do It Yourself Haptics: Part I*” 2007.* IEEE Robotics & Automation Magazine 1070-9932/07 December 2007

[Hecht] David Hecht, Miriam Reiner. 2009. “Sensory dominance in combinations of audio, visual and haptic stimuli” Exp Brain Res (2009) 193:307–314

[Hollins] Hollins M, Bensmaïa SJ, Washburn S. 2001. “ Vibrotactile adaptation impairs discrimination of fine, but not coarse, textures”. Somatosens Mot Res. 2001;18(4):253-62.

[Karplus] Kevin Karplus and Alex Strong. 1983. “Digital Synthesis of Plucked-String and Drum Timbres” Computer Music Journal, Vol. 7, No. 2 (Summer, 1983), pp. 43.

[Kaltenbrunner] 2004. Martin Kaltenbrunner, Sile O’Modhrain, Enrico Costanza “Object Design Considerations for Tangible Musical Interfaces” Proceedings of COST287-ConGAS Symposium on Gesture Interfaces for Multimedia Systems UK 2004.

[Laurienti] Paul J. Laurienti. Robert A. Kraft. Joseph A. Maldjian. Jonathan H. Burdette. Mark T. Wallace. 2004. “Semantic congruence is a critical factor in multisensory behavioral performance.” Exp Brain Res (2004) 158: 405–414.

[Lederman] S. J. Lederman R. L. Klatzky “Haptic perception: A tutorial” Attention, Perception, & Psychophysics 2009, 71 (7), 1439-1459

[Loomis] Jack M Loomis. 1981. “On the Tangibility of Letters and Braille” Perception & Psychophysics 1981, 29 (I), 37-46.

[Mensvoort] K. Van Mensvoort. 2002. “What you see is what you feel, Exploiting the dominance of the visual over the haptic domain to simulate force feedback with cursor displacements”. Proceedings Designing Interactive Systems 2002, ACM Press, 345-348

[Morgenbesser] “Force Shading for Shape Perception in Haptic Virtual Environments” December 1996. Hugh Brian Morgenbesser and Mandayam A. Srinivasan. RLE Technical Report No. 606.. Dept. Of Electrical Engineering, MIT.

[Müller] Alexander Müller, Fabian Hemmert, Götz Wintergerst, Ron Jagodzinski. 2010.“Reflective Haptics: Resistive Force Feedback for Musical Performances with Stylus-Controlled Instruments.” Proc. NIME 2010, Sydney, Australia.

[Oboe 2006] Roberto Oboe. 2006 “A Multi-Instrument, Force-Feedback Keyboard” Computer Music Journal, Vol. 30, No. 3 (Autumn, 2006), pp. 38-52.

[Overholt]Dan Overholt “The Overtone Fiddle: an Actuated Acoustic Instrument”. 2011. Proceedings of the International Conference on New Interfaces for Musical Expression, 30 May - 1 June 2011, Oslo, Norway

[Peters] Deniz Peters. 2013 “Haptic Illusions and Imagined Agency: Felt Resistances in Sonic Experience.” Contemporary Music Review. May 2013

[Pinch 2004] T. J Pinch, Frank Trocco. “Analog Days: The Invention and Impact of the Moog Synthesizer” Harvard University Press, 2004.

[Richard] Christopher Richard and Mark R. Cutkosky. 2002 “Friction modelling and display in haptic applications involving user performance.” Proceedings of the 2002 IEEE International Conference on Robotics 8. Automation. Washington, DC May 2002

[Salisbury] Kenneth Salisbury, Federico Barbagli, Francois Conti. “Haptic Rendering: Introductory Concepts” Stanford Robotics Lab - Stanford University - Stanford, CA, U.S.A.

[Rovan 2000] Joseph Rovan and Vincent Hayward. “Typology of Tactile Sounds and their Synthesis in Gesture-Driven Computer Music Performance.” Trends in Gestural Control of Music 2000.

[Salcudean] S.E. Salcudean and T.D. Vlaar. 1996. “On the Emulation of Stiff Walls and Static Friction with a Magnetically Levitated Input/Output Device”

[Snibbe] “Haptic Techniques for Media Control” 2001. Scott S. Snibbe, Karon E. MacLean, Rob Shaw, Jayne Roderick, William L. Verplank, Mark Scheeff. Proceedings of UIST 2001 Orlando, Florida.

[Spence] Charles Spence. 2012.“Auditory contributions to flavour perception and feeding behaviour.” Physiology & Behavior 107 (2012) 505–515

[Yuan 2008] Bei Yuan Eelke Folmer. “*Blind Hero: Enabling Guitar Hero for the Visually Impaired.”* ASSETS’08, October 13–15, 2008, Halifax, Nova Scotia, Canada.

Additional Literature

The following texts are not cited directly but proved invaluable during the development of this project.

David Byrne. “How Music Works” 2012. Pub. McSweeny’s.

Andy Farnell. “Designing Sound” 2010. Pub. MIT Press

Johannes Kreidler “Loadbang. Programming Electronic Music in PureData.” 2009. Pub. Wolke Verlag.

Jeremy M. Wolfe et. al. “Sensation and Perception” 2nd Edition. 2009. Pub. Sinauer Assosiates.

Miller Puckette. “The Theory and Technique of Electronic Music” 2007. World Scientific Publishing Co.