Haptics in the Air - Exploring Vibrotactile Feedback for Digital Musical Instruments with Open Air Controllers

Håkon Knutzen

Department of Musicology
University of Oslo
Norway

October 2013

Abstract

When playing a traditional musical instrument, the performer often relies on vibration that is produced by the instrument. When perceived through tactile sensing, this can be defined as *vibrotactile feedback*. Since sound in digital musical instruments (DMIs) is not produced by mechanical vibration of its constituent parts, vibrotactile feedback is inherently absent. This means that DMIs are lacking an important feedback modality. DMIs can be played using a wide range of different controllers. *Open air* controllers can make use of motion carried out in open air to control sound. These controllers are particularly prone to the issues related the lack of vibrotactile feedback since they may not have a tangible interface.

In this thesis it was investigated how open air controllers can be augmented with vibrotactile feedback. With basis in relevant theory and previous attempts, two DMI prototypes based on open air control of sound were developed. The prototypes allowed control of musical sound on a high and low level. Open air motion was captured using motion capture technology. In this case, the control surface consisted of a tangible element, such that actuators could be embedded in the controller. It was investigated how vibrotactile feedback can convey musical information. This issue was investigated from both a theoretical and practical approach. The practical approach entailed providing vibrotactile feedback to the fingertips of the performer using signals that were synthesized in musical programming environments. Preliminary results of an informal evaluation of the developed vibrotactile feedback strategies suggest that information on musical parameters such as amplitude and timbre can be conveyed with vibrotactile feedback. While the importance of vibrotactile feedback is stressed in the literature, the preliminary results also show that the developed feedback strategies can be found useful.

Acknowledgments

First of all, I would like to thank my thesis supervisors Tellef Kvifte and Alexander Refsum Jensenius for structured guidance and inspiring feedback throughout the duration of the thesis research. I am grateful to Marcelo Mortensen Wanderley for giving me helpful advice on my thesis research, and for hosting me in the IDMIL at McGill University while I was there as a graduate research trainee.

I would like to thank all the students I met in the IDMIL. They were all very generous and a great inspiration. I would especially like to thank Marcello Giordano for fruitful experimentation with vibrotactile actuators and other gadgets. Joseph Malloch, Stephen Sinclair, Avrum Hollinger, Mark Zadel, Darryl Cameron and Clayton Mamedes for devoting their time to help me with the various issues one may encounter with both hardware and software. Also, Julian Vogels, Mailis Rodrigues, Euler Teixeira and Aaron Krajeski for trying the vibrotactile gadgets I created.

Thanks to everyone in the fourMs/Robin lab for interesting discussions and help with various technological problems. I would like to thank Rolf Inge Godøy for introducing me to interesting facets of music research. Many thanks to my family for support, proofreading, and for providing feedback.

Contents

List of Figures

List of Tables

List of Acronyms

ACROE	Association pour la Création et la Recherche sur les Outils d'Expression
DMI	Digital Musical Instrument
FFT	Fast Fourier Transform
GUI	Graphical User Interface
HCI	Human Computer Interaction
IDMIL	Input Devices and Music Interaction Laboratory
IR	Infrared
MIDI	Musical Instrument Digital Interface
MoCap	Motion Capture
OSC	Open Sound Control
Pd	Puredata

Chapter 1

Introduction

Traditional acoustic instruments are mechanical systems governed by the principles of physics. They consist of membranes, pipes, bars, and strings that are coupled to a resonating body, such as the body of a guitar. When excited, such systems produce audible sound, as well as vibrations that propagate through the instrument. The sound produced by the instrument is undoubtedly the key component that enables the performer to play music. When playing traditional instruments, one is controlling the musical parameters of the sound on a low level. This means that the performer is manipulating the smallest building blocks of what may constitute the sound in the musical context. The performer will, throughout the performance, moderate the playing, such as intonation, based on the audible feedback from the musical instrument.

Audible sound is not the only way in which the performer can get feedback on the way she plays the instrument. Since the performer is controlling the musical instrument intimately, the vibrations propagating through the instrument can be felt as well. Such felt vibrations can be seen as *vibrotactile feedback*. The term *vibrotactile* points to the sensation of perceiving vibration through tactile perception. When playing a musical instrument vibration is perceived through tactile sensing, for instance with the fingers and the lips such as when playing wind instruments.

Askenfelt and Jansson (1992) studied the vibrations found in the double bass, violin, guitar, and piano. They found that the vibration magnitudes were well above the threshold of human tactile perception.

> "[...] informal questioning of professional musicians reveals that musicians seem to benefit from instruments' vibrations for intonation in some situations, in particular in ensemble playing at loud dynamics where it is difficult to monitor one's own instrument" (Askenfelt and Jansson, 1992, p. 347).

The sensing of tactile stimuli is "tightly looped" with the motor system of the human body (Rovan and Hayward, 2000). This means the sensing of tactile stimuli is an efficient feedback modality when playing acoustic instruments. Musicians often rely on kinesthetic cues as well. Such an example might be when a guitarist is navigating on the fretboard using feedback that is obtained when pressing down the fingertips on the strings (Kvifte, 2007).

The term *feedback* points to the fact that such felt vibration can be seen as feedback on the way the performer is playing the instrument, that is, the vibration pattern varies in accordance with the parameters of the musical sound. Thus, one can see the performer and instrument as part of a feedback loop, in which audible, visual and vibrotactile cues are a part.

1.1 Motivation

The main idea of this thesis is to investigate the importance of vibrotactile feedback in open air controllers. Digital musical instruments (DMIs) are widespread today and are being used frequently in many musical contexts. The most known examples are DMIs based on the keyboard paradigm, meaning that they are controlled by a keyboard-like interface. Other DMI designs are also widespread, such as samplers and other devices that can be controlled by using trigger pads. These are however not the only DMIs that have been made. The *New Interfaces for Musical Expression* (NIME) is a series of conferences that, over the last 12 years, has addressed the issues of developing, composing, and performing with existing and novel DMIs.

The reason for focusing on vibrotactile augmentation of DMIs with an open air controllers in this thesis, is the emergence of new open air controllers the last years. The phenomenon of open air controllers, sometimes referred to as "hands-free" controllers, is not a new one (Mulder, 2000). However, the last few years commercial game console controllers such as Microsoft Kinect,[1] the WiiMote,[2] and Playstation Move[3] have been made available to the general public. To mention one example, the Kinect is capable of tracking a user's motion and limbs. Using the position of the performer's limbs, musical sound can be controlled (Bekkedal, 2012). The factors that have made the Kinect so popular are for instance the relatively low price compared to similar devices, as well as the software frameworks that have been made available (Zhang, 2012).

[1] www.xbox.com/KINECT
[2] https://en.wikipedia.org/wiki/Wii_Remote
[3] https://en.wikipedia.org/wiki/PlayStation_Move

While the Kinect or any of the devices mentioned above were not used in this thesis, the popularity of these devices underlines an important point, namely that motion tracking technologies are today more available to the general public, as well as to developers of DMIs. This year, the Leap Motion[4] controller was released. This device can be placed in front of the computer such that hand motion and gestures can be used to control computer programs.

One can still argue that the keyboard paradigm is the most dominant within the family of DMIs. This paradigm is not always suitable for controlling musical sound in all contexts. For instance, the slogan for the *Soundbeam*, which is a DMI based on open air control, is: “The invisible expanding keyboard in space”.[5] While MIDI keyboards are frequently used they can be seen as limiting for controlling DMIs that are not based on the keyboard paradigm. Thus, the keyboard paradigm might be a limiting factor for new DMI designs (O’Modhrain, 2001; Wessel and Wright, 2002).

Similar to Askenfelt and Jansson (1992), Chafe (1993) showed that vibrations on the fingertip that presses the string to fingerboard on a string instrument can be empirically measured using an accelerometer. Inspired by these findings, Chafe synthesized an audio signal that was fed to a voice coil *actuator*, namely a device that produces vibration in accordance with the signal it is being fed with. In this way, vibrotactile feedback was created.

Tactile sensing encapsulates sensing of texture and vibration through the skin, while the term *kinesthetic* deals with sensing of force applied to the body. Both tactile sensing and kinesthetic sensing belongs to the category *haptics*. In this thesis the emphasis is on tactile sensing, more specifically on sensing of vibration sensation in musical contexts through tactile sensing with the hands.

Chafe (1993) emphasized an important issue regarding DMIs, namely that vibrotactile feedback is not present in DMIs unless actuators are integrated in the DMI control surface. This can be seen as an *augmentation* of the DMI controller. The notion of augmentation entails that the interface may still work, even when the vibrotactile feedback is not provided.

Since the inherent vibrations found in acoustic instruments are not present if actuators are not embedded in the design, the DMI might be lacking the perceptual qualities found in an acoustic instrument (Chafe and O’Modhrain, 1996). This is related to the “feeling” of the instrument. As Askenfelt and Jansson (1992) pointed out in the quote above, another argument in favor of haptic augmentation is the lack of feedback on the

[4] www.leapmotion.com
[5] http://www.soundbeam.co.uk

mentioned parameters the performer relies on to play the instrument.

The above mentioned arguments provide a rationale that is in favor of vibrotactile augmentation of DMIs. I am interested in the subject on a personal level both out of sheer curiosity, but also because I believe research on vibrotactile augmentation of DMIs may contribute to the different fields of research and inspire longitudinal use of new DMIs. It is so because I believe haptics may contribute to making DMIs being more interesting to play with.

One proposed approach to deal with this issue is to create vibration using a model of the mechanics of acoustic instruments (Cadoz et al., 2003). While this approach is based on traditional musical instruments, it is perhaps important to stress that control of musical sound with DMIs is not constrained by the fashion of controlling musical sound on a low level, that is, by controlling the smallest building blocks of musical sound. Such building blocks may for instance be individual notes and timbral nuances.

High level control of music is not a new phenomenon, considering the role of conductors in orchestras. Another example of high level control of music was introduced by DJs. They showed that music can be controlled for instance by using analog turntables or other digital technology. A well known example from the field of computer music is the Radio Baton (Mathews, 1991). The Radio Baton is a device that allows the performer to conduct a music program by moving batons in open air. The importance of bringing up the issue of high level control of sound is to emphasize the fact that, in addition to the physical layout of the instrument, a much broader definition for DMIs exist than for traditional instruments. In other words, the term *digital musical instrument* refers to a system that lets the performer control music on several levels (Birnbaum et al., 2005; Malloch et al., 2006).

Novel haptic displays can be used to explore new dimensions of musical expression, learning, composition, and performance. This may entail using vibrotactile stimuli in a learning display for instrument practicing (Giordano and Wanderley, 2011), or composing for the tactile sense by subjecting the listener to spatially laid out vibrotactile stimuli by using a full body suit (Gunther and O'Modhrain, 2003). Another example is *sensory substitution*. With sensory substitution vibrotactile feedback can enable people that cannot hear or see to participate in creating music (Egloff, 2011). DMIs with open air controllers are interesting with regards to haptics, since the performer may not be using a physical interface to navigate and control the musical sound. Augmenting the controller with vibrotactile feedback gives the opportunity to study the interaction with the DMI — with and without any vibrotactile feedback.

Most people deal with haptics in everyday life (Grunwald, 2008). Consider for instance interaction with mobile phones. Here vibrations can provide notifications e.g. on received calls. Haptics is a growing field that embraces many fields of research such as human computer interaction (HCI), engineering, psychology and music. Often these fields combine in multidisciplinary approaches. As I see it, one may argue that research on haptics within individual disciplines can be beneficial for the larger field of research on haptics.

1.2 Research Questions and Problem Domain

This thesis will seek to answer the following main research question:

- How can vibrotactile augmentation be implemented in a DMI design with an open air music controller?

Within this question, it is important to emphasize the aspect of music control, since many of the controllers used in DMIs are generic controllers used to control a wide range computer programs. Derived from the main research questions are two sub-questions:

1. What musical information can be conveyed with vibrotactile feedback?
2. Can vibrotactile augmentation be useful in the context of playing the given DMI?

Answering these questions involves a multidisciplinary approach. I will here introduce and explain the different fields of research and how they contribute to the answers of the questions: (1) Embodied music cognition; (2) Computer music; (3) Human computer interaction (HCI); and (4) Motion capture.

An offspring of systematic musicology is *embodied music cognition* (Leman, 2008; Godøy and Leman, 2010). This direction builds upon ideas of *embodied cognition* (Shapiro, 2011) to investigate and explain music perception, cognition, and musical practice with the basis in the human body. Thus, I wish to contextualize musical practice with an open air controller and vibrotactile feedback in the perspective of embodied cognition. I find this important because perception of vibrotactile sensations adds to the sense modalities involved in multimodal perception when playing music. Computer music is the field dealing with both realtime and non-realtime control of computers for creating music (Roads, 1996). HCI is the field that deals with human control of computers on a general level (Dourish, 2001; O'Sullivan and Igoe, 2004). Thus, central aspects of HCI also relates to computer music and DMIs. Motion capture (MoCap) involves different techniques for tracking and recording motion. While MoCap can be used for a wide range of purposes, some examples of use in musical contexts can be found in (Mamedes et al., 2013; Dobrian and Bevilacqua, 2003; Nymoen et al., 2011).

Explaining what the term *music* entails, is too much of an elaborate task for this thesis since the discussion of the term is a long one (Nettl, 2013). Acknowledging the fact that the study of music and its many facets involves a variety of disciplines and theories, I will not propose a strict definition of the term. Rather, I will adopt a flexible approach to the term in which musical control of sound is stressed. More specifically, in the context of using an open air controller to control sound, and how vibrotactile stimuli can provide feedback on musical parameters in the given context.

1.3 Research Design

To answer the research questions, focus will be on the constituent parts, the conception and the development of DMIs, as well as how vibrotactile integration may be used to augment DMI designs. First, I will provide an overview of relevant DMI designs and literature. Then, the process of the exploration and the development of the two constructed DMI prototypes and the vibrotactile strategies is explained in detail. The DMI prototypes were used to exemplify how vibrotactile feedback could be integrated in an open air controller, as well as how vibrotactile signals can covey musical information. The controller for the two DMI prototypes was based on optical MoCap technology. Actuators were used to provide vibrotactile feedback. Using musical programming environments, sound synthesis and synthesis of vibrotactile signals were programmed.

In other words, this is a specific approach to address the research questions. Also, considering the involved fields of research in the thesis, the emphasis is on the musical aspects of DMI construction and vibrotactile feedback. However, this involves a technical explanation of hardware, programming and sound synthesis.

Considering the wide range of available musical instruments and how they are in contact with different body parts, the sensing of tactile stimuli may involve several parts of the body. As a constraint, this thesis will primarily be dealing with vibrotactile stimulation of hands, and more specifically the fingertips. The hands are very sensitive to vibrotactile stimuli, and very involved in musical performance (Verrillo, 1992). This means that an elaborate explanation of vibrotactile feedback for other parts of the body, as well as kinesthetic feedback, will not be provided in detail.

1.4 Structure of the Thesis

Chapter 2 provides an overview of the necessary theory for contextualizing the research of the thesis. Here vibrotactile perception as well as multimodal perception, motion, action, and gesture are explained in the scope of physiology and embodied cognition. The theory behind DMIs is explained, and terminology within the research on DMIs is presented. Motion capture technology is explained alongside an overview of some DMIs with open air controllers. An overview of some musical controllers with haptic augmentation is also provided.

Chapter 3 presents the process of choosing hardware and software for the DMI prototypes. This entails providing an explanation of the rationale behind the selection process, as well as the technical implementation of the prototypes with respect to both programming and hardware. Before ending up with two DMI prototypes, an exploration of the given hardware and software is presented.

In Chapter 4, an informal evaluation of the vibrotactile strategies explained in Chapter 3 is presented and discussed. The participants in the informal study were five graduate students from the IDMIL (Input Devices and Music Interaction Laboratory) at McGill University. The evaluation seeks to investigate the functionality of the vibrotactile augmentation of the two DMI prototypes with respect to the research questions. The whole thesis content is then summarized and discussed in Chapter 5.

The appendix contains SuperCollider code and screenshots of Puredata and Max MSP patches that were used in the exploration and implementation of the DMI prototypes. Matlab code for analysis of the vibrotactile signals as well as the a part of the hardware used in the implementation is also provided in the appendix.

Chapter 2

Background

This chapter presents an overview of the theory needed to address the questions presented in Chapter 1. As explained, this thesis has a multidisciplinary approach. Therefore, related theory and fields of research will be presented in this chapter. The terms *motion*, *action*, and *gestures* are explained in the scope of embodied music cognition. The terms *multimodal perception* and *vibrotactile perception* are then explained. This serves as a theoretical background for describing interaction with DMIs and vibrotactile feedback. The theory of DMIs is explained along with an overview of DMI designs with open air controllers, as well as an overview of haptic displays for DMIs.

2.1 Embodied Music Cognition

I have chosen to present the thesis content with respect to embodied cognition when dealing with DMIs and vibrotactile feedback. Embodied cognition is a direction in psychology that has contested the traditional idea of cognition (Shapiro, 2011). Sensory information is obtained through the sensing organs (perception). Traditionally, one has seen *cognition* as a process where sensory information is processed and interpreted to form basis for further action. The processing of the sensory information and the processes of the cognitive domain have been regarded as belonging to the mental sphere exclusively. *Embodied* cognition involves the human corpus in the cognitive process. In the realm of embodied cognition there is no dichotomy between the corporeal perception and mentally based cognition.

Embodied *music* cognition can be seen through the same scope (Leman, 2008). In traditional western thinking, the cognition of music has been seen as belonging to the sphere of mental processes that take place in the human mind. Embodied music cognition can be seen as an attempt to bridge the mental processes related to music perception and cognition, and musical practice and human action. Compared to traditional musicology that only focuses on the musical scores or the musical sound, embodied music cognition is a contrasting approach.

2.1.1 Motion, Action, and Gestures

The terms *motion*, *movement*, and *gesture* are in music research with basis in embodied cognition (Godøy and Leman, 2010). Gesture points to the meaning bearing element of motion:

> "[...] gesture somehow blurs the distinction between movement and meaning. *Movement* denotes the physical displacement of an object in space, whereas *meaning* denotes the mental activation of an experience. The notion of gesture somehow covers both aspects and therefore bypasses the Cartesian divide between matter and mind." (Jensenius et al., 2010, p. 13)

The quote above points to the significance of gesture in embodied cognition. Although the term *movement* is frequently used interchangeably with *motion*, I will choose to use the term *motion* to avoid confusion with the movements of musical pieces. One can argue that musical performance and interaction might entail motion that is not necessarily considered to be gestures. Any moving object might produce sound when striking for example a piano key. With no mental awareness of the motion that results in sound, either from the acting or perceiving subject, this can be seen as an *action*. The gestural elements may be absent, i.e. that no element of meaning is conveyed through the motion.

As explained by Jensenius et al. (2010), the discussion of gestures is long. I will not elaborate on this discussion, but I believe a distinction between gesture, motion, and action is important. I will only use the term gesture when dealing with motion that conveys content that can be considered meaningful by either the person that executes the gesture, or the person perceiving the executed gesture. I will use action when referring to events such as sound production. Note that this does not mean that the two terms are mutually exclusive.

Jensenius et al. (2010) provided a categorization of motion associated with music. With basis in this categorization, and with respect to the explained distinction between motion and gesture, I will here propose a classification of motion and actions related to music. *Selective* actions encapsulates sound *producing* actions and sound *modifying* actions. A sound *producing* action denotes actions used to produce sound, such as when hitting a piano key. A sound *modifying* action might be that of pressing the sustain pedal on the piano while playing. Sound *accompanying* motion encapsulates motion that is not directly linked to sound production. Belonging to this category are *ancillary* gestures, *sound facilitating* motion, *entrainment*, and dance. Such motion may be used to trace the phrases of a music, that is, one can regard this as perceiving and rendering music in an embodied manner (Godøy et al., 2006). This means that human action is central in embodied music cognition.

In musical practice, ancillary gestures may also be directly linked to the course of the musical material being played. This may be an expressive outlet for the performer (Nusseck and Wanderley, 2009; Wanderley et al., 2005). In the case of clarinet performance it has become evident that ancillary gestures also affect the timbre of the sounds emitted by the clarinet. This is because the continuous change in orientation of the clarinet bell also changes the reflection pattern of the sound coming from the clarinet. In turn, this results in a comb filtering effect (i.e. timbral changes) of the clarinet sound (Wanderley and Depalle, 2004). In this case, ancillary gestures are also sound modifying actions. Sound facilitating motion denotes motion that helps the performer execute sound producing and sound modifying actions. Entrainment denotes motion such as tapping the foot along with the music, or other motion which may occur when a person gets "carried away" with the music. In musical practice, these different forms of motion, actions and gestures occur both sequentially and simultaneously, meaning that they are not always separable.

With respect to embodied music cognition and DMIs, it is useful to have a distinction between action-sound *couplings* and action-sound *relationships*. While all action-sound couplings can also be action-sound relationships, it is not so for the opposite. According to Jensenius (2013), action-sound couplings only holds for the relationships between a sound producing action and a mechanical system which results in sound. This is found in acoustic instruments. Such a robust coupling is not found in DMIs, since similar mechanical couplings are not inherent. Instead, the action-sound relationships in DMIs are results of arbitrary couplings between digital and electronic signals and components.

Having a typology and understanding of musical motion and gestures is not only useful for research on music perception, cognition, and practice within embodied cognition. An understanding of musical motion and gestures might be very helpful when developing new DMIs. It is so because one can more easily target specific body motion or gestures in the design process, both with regards to the controller surface layout, choice of sensors, and mapping. It is also useful to have a typology of motion, gestures and actions when studying the interaction with DMIs.

2.2 Haptics

This section explains key terms and theory needed for understanding haptics and tactile perception. Egloff (2011) pointed out that the term *haptic* involves the sense of touch and is related to the Greek word "haptein", which means "to touch", while the word "tactile" stems from the latin word "tactilis", which is the past tense of "tangere" meaning "to touch". Rizun et al. (2006) explains that *haptic* "from the Greek haphe, means pertaining to the sense of touch"(p. 343). Tactile sensing differs from haptic sensing since it is placed within the category *somethesis* in the literature of psychology and physiology (Sinclair, 2012, p. 3).

Also belonging to somethesis are *proprioception* and *kinesthesis*. Proprioception entails the sensing of the state of the whole body through *cutaneous*, *kinesthetic*, and *vestibular* perception. *Cutaneous* refers to the perception through the skin, *kinesthetic* sensing refers to perception of motion, and *vestibular* sensing to the acceleration, deceleration, and position of the head (Oakley et al., 2000).

Like Sinclair (2012) pointed out, the distinction between haptic and tactile lies in the difference between *active* and *passive* sensing, haptic belonging to the former, while somethesis belongs to the latter. However, he also points to the fact that lately the term haptic is frequently being used to denote both passive and active perception. I will make no distinction between active and passive experience of haptic stimuli in this thesis. Instead, haptics will be used as an umbrella term for both tactile and kinesthetic perception (Oakley et al., 2000). The former is a focal point in explaining the thesis problems. Tactile sensing deals with the sensing of stimuli through the skin. This can be the sensing of vibration, texture of materials, temperature and pain.

2.2.1 Tactile Sensing

The distinction *vibro*tactile is useful when speaking of vibration stimuli perceived through the skin, since other sensations such as temperature, pain, and kinesthetic sensing are not taken into account. I will in this section explain vibration perception in more detail. This means that I will not focus extensively on kinesthetic perception which entails perception of force applied to joints or muscles, nor will I explain proprioception, or sensing of temperature and pain in more detail.

While vibrotactile perception in general entails perception with the whole body, an

Table 2.1 The four mechanoreceptors in the glabrous skin and their characteristics. The table is based on Table 1 in Choi and Kuchenbecker (2012). Note that the range of perceivable frequencies may vary in the literature.

Mechanoreceptor	Neural Channel	Frequency Range (Hz)	Spatial Resolution
Meissner corpuscle	FA I	3–100	High
Merkel disk	SA I	< 5	High
Pacinian corpuscle	FA II	10–500	Low
Ruffini ending	SA II	15–400	Low

elaborate explanation of the physiology of such a subject is beyond the scope of this thesis. The hands are very important body parts for tactile perception and interaction (Verrillo, 1992). This is due to the fact that there are around 17000 mechanoreceptors in the hand (Vallbo and Johansson, 1984).

Vibration perception with tactile perception in the hands is mainly attributed to the four different *mechanoreceptors* located in the glabrous (hairless) skin (Halata and Baumann, 2008). The four kinds of mechanoreceptors are *Meissner corpuscles*, *Merkel disks*, *Pacinian corpuscles*, and *Ruffini endings* (Table 2.1). These can be subdivided into two categories (Vallbo and Johansson, 1984). The first category is Slow Adapting, abbreviated SA I and SA II. The second is Fast Adapting, abbreviated FA I and FA II.

SA I and SA II are labeled slow adapting since they both respond to dynamic and static stimuli, while FA I and FA II are called fast adapting since they only respond to dynamic stimuli. Adaptation can be considered to be important since repeated exposure to stimuli may yield fast adapting mechanorecetors to be less sensitive to further stimulation. This means that although fast adapting mechanoreceptors are being exposed repeatedly to stimuli, they do not trigger neural responses throughout the exposure. In practice, this may be a an important consideration when designing vibrotactile stimuli, since one may wish to keep the desensitization of the FA receptors to a minimum.

SA I and FA I have such characteristics that they allow perception of high spatial resolution, while SA II and FA II have low spatial resolution. A high spatial resolution allows more accurate localization of stimuli. The mechanorecetors contribute to perception of different stimuli. Merkel disks respond to fine details, Meissner corpuscles to “flutter”, Ruffini endings to stretch and pacinian corpuscles to vibration (Choi and Kuchenbecker, 2012). When comparing tactile perception to auditory perception one can therefore see that the two are very different, since tactile perception involves several sensing organs. Although the auditory system is composed of different components, the individual frequency bands are not assigned to different organs (Rossing et al., 2002).

The neural impulses sent from the mechanorecetors can be measured individually (Goodwin and Wheat, 2008). However, stimuli will naturally excite different kinds of mechanoreceptors simultaneously. Thus, the interplay of the stimuli perceived by all the mechanorecetors combined makes it a complicated matter. Although one can distinguish between different sense modalities (tactile sensing being one modality) within which different organs are responsible for the different parts of the sensing, it is important to acknowledge the fact that one most often perceive with different sensing modalities simultaneously. This is called *multimodal* perception. The signal one can therefore perceive through different sensing modalities may be redundant and nonredundant (Partan and Marler, 1999). The distinction lies in whether or not the signals represent the same phenomenon. Redundant signals may result in stimuli being perceived equally intense, while they may also result in increased intensity. Nonredundant signals may be perceived as independent or emergent, or to modulate and dominate the other perceived signals. The benefit of multimodal perception is, among others, reduced ambiguity, increased performance, precise judgment, and enhanced detection (Helbig and Ernst, 2008). Thus, musical vibrotactile feedback can contribute with either redundant or nonredundant stimuli.

2.3 Digital Musical Instruments

In acoustic instruments the sound *generator* and the sound *controller* are connected mechanically. The sound generator is for instance a string coupled to a resonating body, while the controller is for instance the fretboard of a guitar or the keyboard of a piano (Rossing et al., 2002). This coupling lays down many of the acoustic and mechanical properties of the instrument. Thus, complex physical interference occurs between the latter and the former. This means that sound producing and sound modifying actions interfere as well. As an example of this, consider how modification actions such as pressing down the strings on the guitar also directly interferes with a part of the sound generator, namely the string. Pressing down or pulling the finger off the fretboard rapidly may excite the string.

The mechanical system that constitutes an acoustic instrument produces audible sound and vibration that can be perceived as vibrotactile feedback. In this respect DMIs differ from acoustic musical instruments. The connection between the sound generator and the sound controller in DMIs is not governed by principles of physics and acoustics (Jordà, 2004). It is so because the sound controller consists of sensors that send electric signals, that are sampled and mapped to a digital sound producing algorithm. In other words, the inherent and complex mechanical coupling as found in acoustic instruments does not exist in DMIs.

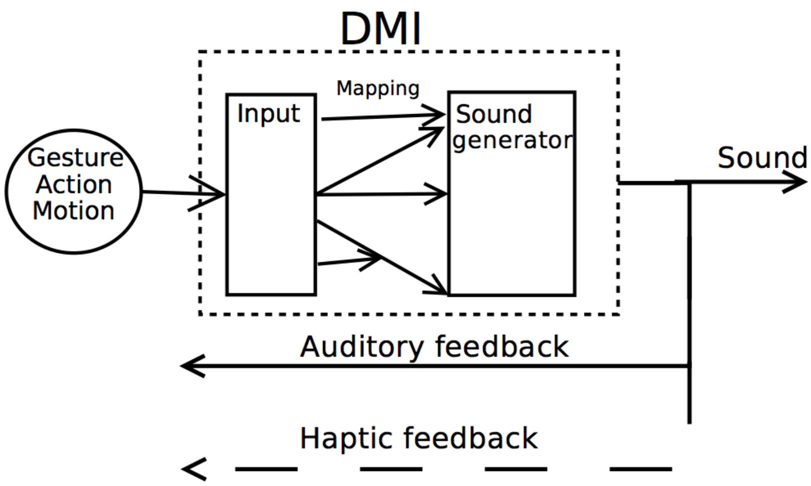


Figure 2.1 Architecture of DMIs.

The constituent parts of a DMI are typically different kinds of sensors or buttons; a computer capable of performing sound synthesis; and a device that can produce physical sound (e.g. loudspeakers). The way in which these constituent parts are coupled is arbitrary. Thus, the notion of DMIs is therefore broader than for acoustic instruments, since it encapsulates all the individual components needed for controlling and producing sound. The components are not necessarily contained within one unit, as in the case of acoustic instruments. A variety of sensors can be used to obtain input from a human performer (Miranda and Wanderley, 2006).

2.4 Sound Generator

Term *sound generator* encapsulates all the constituent pieces of the DMI that are used to create sound. This means the algorithm used to synthesize a signal, the digital to analog converter (DAC) that converts the digital signal to an analog signal, and the loudspeakers that eventually turn the electric signal into physical sound waves. There exists a wide range of digital audio workstations (DAWs) with various software synths that are capable of both playing back stored sounds, or synthesizing sound. However, other environments offer much more flexibility when wanting a customizeable approach. This may be important when constructing DMIs. For such purposes different control and audio programming environments can be used (Collins, 2010, p. 33). I will here present the programming environments used in the implementation in this thesis:

- Max MSP
- Puredata (Pd)
- SuperCollider

Max Msp is a visual programming environment that can be used to synthesize and control synthesis of sound. Different objects can be connected to each other using virtual patch cords. Each of these objects can typically perform functions ranging from simple ones, such as addition, to more complex ones such as fast Fourier transforms (FFTs). When connected together, complex programs can be formed. The environment in which all these objects are coupled to each other with virtual cords is called a *patch*.

Puredata (Pd) is the open source sibling of Max Msp (Puckette, 2007). This means that it is free to use and the source code is public. On the other hand, Max Msp is commercial and maintained by the company Cycling74.[1] Both Max Msp and Pd offer flexibility with respect to rapid prototyping, not only because connections between the objects can be created and destroyed easily, but also because both environments run in real time. This means that audio and control rate signals can flow continuously while the patch is being edited. Sensor input can be accessed via MIDI, serial port and open sound control (OSC).[2]

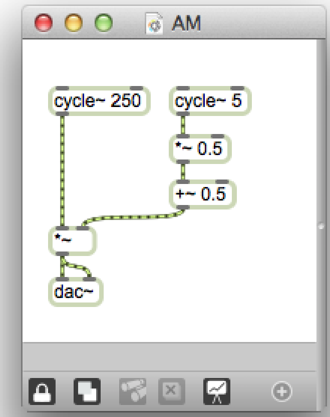
Another environment that also offers highly customizable sound synthesis and control is SuperCollider. This environment is text based, efficient, and operates in real time (Wilson et al., 2011). Synth definitions (called `SynthDef` in SuperCollider code) can be used to store synthesis algorithms. These synth definitions can be called and retrieved elsewhere in the code. This environment also operates with MIDI, serial, and OSC communication. The explanation on the sound generator in this thesis is mainly related to the synthesis and control of musical signals using these mentioned environments.

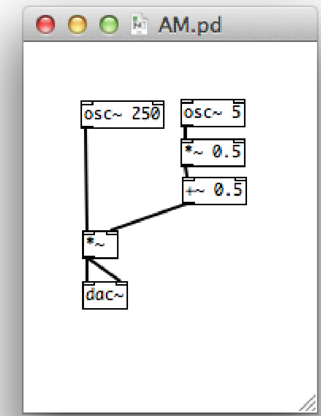
2.5 Controllers

Controllers for DMIs are often referred to as input devices or gestural controllers in the literature. Controllers are necessary for bridging the human and the computer.

[1] http://cycling74.com

[2] For more information on OSC see (Wright, 2005)






```
{SinOsc.ar(250)*SinOsc.ar(5,mul:0.5,add:0.5)}.play;
```

Figure 2.2 Examples of unipolar amplitude modulation of a sinusoidal signal in a Max MSP patch (top), Pd patch (middle) and SuperCollider code (bottom).

This denotes a more generic notion of a controller, i.e. a device that translates human motion, action, and gestures into signals that can be used to control computers. The field dealing with such problems is known as human computer interaction (HCI). Most of the input devices and sensors used in DMIs are ones one would normally use in more general applications as well. Thus, many of the problems related to DMIs have been addressed from an HCI perspective. While the HCI aspect of DMIs is not something I will elaborate on extensively, some explanation is needed.

Miranda and Wanderley (2006, p. 20) distinguishes between the following when speaking of controllers for musical purposes:

- Augmented musical instruments.
- Instrument-like gestural controllers.
- Instruments-inspired gestural controllers.
- Alternate gestural controllers.

Augmented musical instruments denotes existing or traditional instruments that for instance have sensors attached to them (Machover and Chung, 1989; Thibodeau and Wanderley, 2013). Instrument-like controllers have a control surface that closely resembles existing musical instruments. Instrument-inspired controllers make use of principles found in existing instruments, but do not necessarily resemble the instrument itself. Alternate controllers relate to more radical and abstract designs that might neither resemble the appearance nor the behavior of existing or traditional instruments.

A sound controller does not need to offer a physical control surface that the performer touches. Rovan and Hayward (2000) call such controllers *open air* controllers. Thus, such controllers belong to the category alternate controllers. Given the explanation of motion, action, and gestures, I will refrain from appending *gestural* to the term controller.

2.5.1 Open Air Controllers

The oldest and perhaps most obvious example of a musical instrument controlled by open air motion is the Theremin. Although this instrument is not a DMI in its purest form, the sound controller itself can be considered an open air controller. Invented in 1919 by Lev Sergeyevich Termen (also known as Léon Theremin), the Theremin uses two capacitive sensing antennas that are mapped to the amplitude and frequency control parameters of an analog synthesizer (Paradiso and Gershenfeld, 1997). This offers continuous and accurate control of the mentioned parameters using open air hand motion. Since pitch is controlled continuously, accurate pitch localization is difficult. If wanting to use the antennas for controlling other parameters, the control voltage output can be routed to other analog equipment, or be sampled by a microcontroller.

There exists several music controllers that were developed to utilize open air motion. The Buchla Lightning was developed by Don Buchla.[3] With this controller, the performer holds one infrared (IR) emitting stick in each hand. The IR light is picked up by a sensor placed in front of the performer. This way the motion of the performer is detected and translated to MIDI. Another well known open air controller, invented by Max Mathews, is the Radio Baton (Mathews, 1991). This device is sometimes also referred to as the Radio Drum. The controller senses the motion of radio frequency emitting batons held by the performer using an antenna array of radio receivers. The Soundbeam uses ultrasound to measure the distance between the device and the performer. Here the measured distance is processed and translated to MIDI messages. The Soundbeam has been used in music therapy. The above mentioned are only some of the many open music air controllers that exist. I can therefore not go into detail on all of them.

2.5.2 Motion Capture

There exist different kinds of technology that may be used in open air musical controllers:

- **Optical Based**
  - Marker Based
    - Infrared
  - Markerless
    - Infrared
    - Video/Computer vision
- **Sensor Based**
  - Physiological
    - Electroencephalography (EEG)
    - Electromyography (EMG)

[3] http://buchla.com

- Ultrasound
- Magnetic
- Infrared
- Mechanical
- Electric field
- Inertial
- Capacitive

Note that some systems will make use of multiple technologies. In the overview above I have made a main distinction between *optical based* and *sensor based* systems.

Different kinds of technology in MoCap systems are: (1) mechanical MoCap using exoskeletons; (2) inertial based MoCap using accelerometers; (3) IR marker based MoCap; and (4) video or computer vision based MoCap. The first two types of sensing technology will not be explained in detail in the thesis, since the focus will remain on optical based MoCap, that is, the latter two categories. The emphasis will be on marker based MoCap systems. This is because an IR marker based system was used in the implementation in Chapter 3.

In marker based MoCap, the markers can either be active or passive. The former means that the markers themselves emit IR light, while the latter means that the markers simply reflect the IR light emitted by the cameras. With the Vicon,[4] Qualisys,[5] and Optitrack[6] systems, reflective passive markers are attached to the limbs of a subject. Cameras are placed around the subject, and IR light is projected on the scene within which the subject is located. The IR light reflected off the markers is picked up by the cameras. For accurate and precise measurements using IR marker based MoCap other reflective devices in the room should be restricted to a minimum.

Organic Motion,[7] Leap Motion,[8] and Microsoft Kinect are examples of optical based MoCap that can sense motion without the use of passive or reflecting markers. With the Kinect, the limbs of a subject can then be detected and tracked using proprietary algorithms (Zhang, 2012). The Leap Motion can detect and track the hands and fingers of a subject using a form of optical sensing possibly in combination with sensor based technology (Hodson, 2013). Like with IR marker based MoCap, Organic Motion gets

[4] http://www.vicon.com
[5] http://www.qualisys.com
[6] http://www.naturalpoint.com/optitrack
[7] http://www.organicmotion.com
[8] https://www.leapmotion.com

its input from multiple cameras placed around the scene within which the subject is located. This system relies on computer vision techniques.

Once the marker based systems are calibrated properly and markers are picked up by the cameras, the positions within the 3D coordinate system can be obtained. Within the system software, markers can be linked together to form dynamic or rigid models allowing tracking of dynamic skeletons and rigid bodies. Depending on the motion, the camera frame rate can be adjusted. A higher frame rate means rapid motion can be captured accurately. More data will then be recorded which takes more space and effects computational load when processing the data.

IR marker based MoCap systems, such as Vicon, Qualisys, and Optitrack, demand that the subject of capture is wearing a fixed marker configuration inside a designated space. This is because the cameras need to be placed around the subject and calibrated. It may be difficult to use optical based MoCap for DMIs like the Dance Jockey (de Quay et al., 2011). It is so since the stage environment in musical performances may disturb the optical sensing due to reflective objects and changing lighting conditions. Thus, the Dance Jockey bypasses this problem by using inertial based MoCap, which does not rely on optical based sensing. IR marker based systems therefore impose limitations when wanting to capture a musical performance within the performer's natural environment. However, the benefit of using such systems is the accuracy and precision they offer, also with respect to absolute position.

2.6 DMI Designs with MoCap Technology

Table 2.2 Some DMIs where MoCap systems are used in the controller design.

Application	Controller	Reference
Dance Jockey	Xsens	de Quay et al. (2011)
Motion Capture Music	Vicon	Dobrian and Bevilacqua (2003)
Audio Visual Installation	Vicon	Mamedes et al. (2013)
Control of spatialized sound	Inertial/Magnetic	Schacher (2007)
SoundCloud	Kinect/Vicon	Martin (2011)
SoundSaber	OptiTrack/Qualisys	Nymoen et al. (2011)
VMI	Inertial	Mulder (2000)

Table 2.2 contains an overview of some DMIs with MoCap technology. Some of these examples will be presented here. The Dance Jockey utilizes the Xsens inertial MoCap system to capture motion that is mapped to control of sound (de Quay et al.,

2011). The Dance Jockey uses a full-body MoCap system to control prerecorded sound much like a DJ would. Schacher (2007) presented an attempt to control spatialized sound using inertial based MoCap. Similarly, Mulder (2000) presented the notion of a virtual musical instrument (VMI), which entails a control of musical sound in a virtual environment.

Several attempts using marker based MoCap for DMIs exist as well. Dobrian and Bevilacqua (2003) proposed an approach to use a Vicon IR marker based MoCap system to control musical sound. Nymoen et al. (2011) presented how tracking of a rigid object held in the hands of the performer could be used to track the motion. The motion was in turn used to control sound similar the sound of lightsabers in Star Wars. Mamedes et al. (2013) extracted gestures from a performer's motion with basis in Laban theory. The same MoCap system used in this example was also used in Chapter 3. Martin (2011) used open air motion to control concatenative sound synthesis. While non of these particular DMI approaches were pursued in the implementation in Chapter 3, they illustrate the wide applicability of open air control of sound.

2.7 Mapping

In the context of DMIs, *mapping* denotes the coupling of sensor input and parameters of the sound generator. As explained, in acoustic instruments the mapping is determined by the way the instrument is put together mechanically. Inherently, there is no such predefined mapping in DMIs. The issue of mapping in DMIs has been discussed extensively.

The way in which the couplings are set up is often referred to as the mapping *strategy*. *One-to-one* mappings mean that one sensor input is coupled to one synthesis parameter, such as the pitch of an oscillator. One can also implement *one-to-many* or *many-to-one* mapping strategies. The former means one sensor input is coupled to many synthesis parameters, while the latter entails that many sensor inputs are coupled to one synthesis parameter. In most acoustic instruments, the mappings are cross coupled, meaning that several parameters may be controlled by e.g. change in wind pressure (Kvifte, 2008). Therefore, the importance of the mapping strategy is not limited to being merely a description of such couplings. As described by for example Hunt and Kirk (2000), the mapping strategy can radically alter the performer's experience with the behavior of the DMI.

Mappings might be both *implicit* or *explicit* (Hunt and Wanderley, 2002). With explicit mapping, the performer or DMI designer has made couplings between the parameters. With implicit mapping, the computer determines the couplings and operations on the signals based on a machine learning algorithm. An example of tools for creating implicit mapping, is the MnM toolbox for Max MSP from IRCAM (Bevilacqua et al., 2005). Mapping strategies might have multiple layers. One layer might have abstract parameters such as "brightness", while the abstract parameters might in turn be mapped in various fashions to low level features of the sound producing algorithm (Hunt et al., 2003). That way one-to-one mappings between the control parameters and the abstract layer might still result in a complex mapping overall. Also given that the abstract layer is already pre-defined, the user does not have to make couplings to raw synthesis parameters. This can be an advantage if the user does not have in-depth experience with the synthesis algorithm.

DMIs can be *model*-, *rule*-, or *skill*-based (Malloch et al., 2006). Here model-based performance means high level control of musical events, such as live coding and playing back larger segments of musical sound. Rule-based performance means control of lower level musical (relative to model-based performance), such as live sequencing. Skill-based denotes the performance mode that deals with control of the lowest level of musical events such as individual notes and timbral nuances. This is most commonly found in the way one plays traditional acoustic instruments. These distinctions point to another important point of DMIs, namely that DMIs can deal with several different levels of musical manipulation of sound. Given this notion, I choose to keep a broad definition of DMIs instead of limiting the term to dealing with devices that can control musical sound on a low level such as in traditional musical instruments.

2.7.1 Libmapper

In addition to the MnM toolbox, there exist other toolboxes for creating mappings in DMIs, for example Steim's *Junxion*[9] and the *HID* toolkit for Pd (Steiner, 2006). Another example is *libmapper*. The libmapper[10] tool has been developed in the IDMIL at McGill University (Malloch et al., 2013). It offers a flexible way of processing data from input devices as well as creating, destroying and saving mappings between *sources* and *destinations*. OSC is used to send data between the sources and destinations.

A libmapper *device* may have multiple inputs and outputs, each of them listed as individual destinations and sources. Sources are typically input from individual sensors, while destinations might typically be control parameters of a synth. Values can easily

[9] http://steim.org/product/junxion/
[10] http://libmapper.github.io

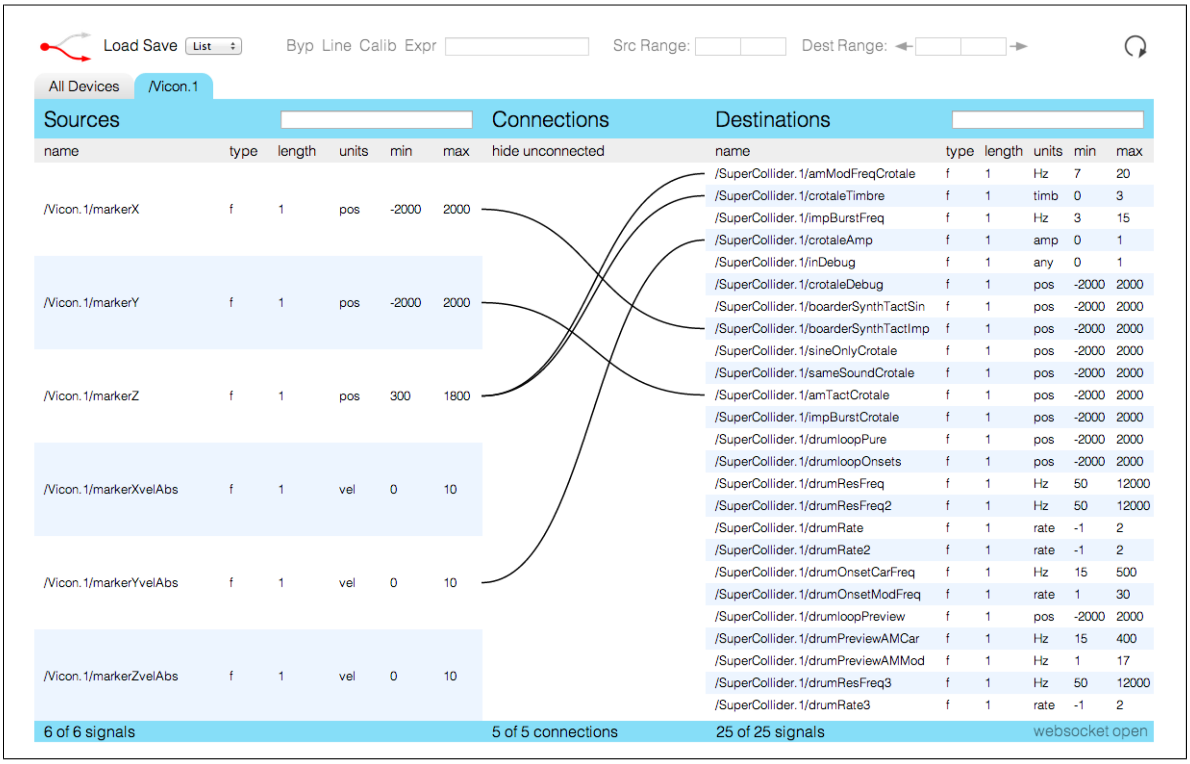


Figure 2.3 Screenshot of the libmapper GUI with an example mapping between sources (left) and destinations (right).

be processed using a built-in expression function. This can be used for scaling and for basic filtering. With the libmapper GUI[11] different mappings can be created and stored using virtual patch cords between the input device outputs and the mapping layer inputs (see Figure 2.3). A recent add-on to the libmapper GUI also offers a grid view of the mapping. Together these views offer a simplified and quick way of creating and editing mapping strategies. Currently there exist libmapper support for Java, C/C++, Python, SuperCollider,[12] Max MSP and Pd.

2.8 Vibrotactile Feedback in DMIs

In the introduction I explained how performers may sense and obtain information on how the instrument is being played through haptic sensing. Kvifte (2007) pointed out that the importance of haptic sensing when navigating on the fretboard on the guitar. Askenfelt and Jansson (1992) pointed out that in certain cases when playing in ensembles, string players could benefit from tactile feedback. The influence of vibrotactile

[11] https://github.com/mysteryDate/webmapper
[12] https://github.com/mzadel/libmapper-sc

feedback on the feeling of the musical instrument was stressed in (Chafe, 1993; Chafe and O'Modhrain, 1996).

In this section I will provide a more thorough explanation of theory related to vibrotactile feedback in DMI, and present some examples of previous attempts of haptic augmentation in DMIs. The architecture of DMIs is illustrated in Figure 2.1 on page 15. A consequence of the fact that the sound generator and the controller is not coupled mechanically is that haptic feedback is not provided inherently. To enable the haptic channel in DMIs, one can embed actuators in the design or (dashed line in the figure). Another approach is to let the material of which the DMI is made of provide for example kinesthetic feedback as a result of the physical attributes of the material (Malloch et al., 2011; Morris et al., 2004). Passive feedback will not be dealt with in this thesis.

Giordano and Wanderley (2013) provided an overview of musical parameters that can be conveyed through vibrotactile feedback. I will in the following elaborate on this overview.

- Temporal Domain:
  - Pitch and Amplitude
  - Rhythm
  - Roughness
  - Timbre
- Spatial Domain:
  - Acuity, pattern recognition and numerosity
  - Tactile illusions
  - Attention

2.8.1 Temporal Domain

Pitch and Amplitude. The most sensitive range of frequencies in vibrotactile stimuli perception is usually said to be between 40–1000 Hz (Verrillo, 1992). The bandwidth of the vibrotactile frequency perception is therefore less than the bandwidth of audible frequency perception. In auditory perception, the *Fletcher-Munson* curve tells us that humans do not perceive all frequencies equally loud (Mathews, 2001). Much in the same manner, we do not perceive vibrotactile stimuli consisting of different frequency content with equal intensity. In fact, this curve shares similarities with the Fletcher-Munson curve in that there is a particular area of frequencies that are perceived more intensely than the other frequencies. In the curve found in Verrillo (1992) there is a "dip" around 250 Hz, meaning that stimuli with frequency in that particular region are perceived more intensely than other frequencies with the same magnitude. The stimuli magnitude (skin displacement) as well as stimuli frequency, are therefore related to perceived vibrotactile intensity. Similarly, the "dip" in the Fletcher-Munson curve is found at around 3–4 kHz.

The just noticeable difference (JND) tells us about the human ability to distinguish stimuli apart from each other. This is often revealed through psychoacoustic experiments. The JND with regards to frequency discrimination in tactile sensing is reported to be around 18% (Pongrac, 2008) to 30% (Goff, 1967). By comparison, the JND in auditory perception is reported to be around 0.5–3% (Loy, 2006, p. 162). This means that the difference between comparable stimuli must be significantly higher in tactile sensing than in auditory sensing for the subject to perceive a difference. Various studies have addressed the issue of sensing pitch through haptic perception. For example, Rovan and Hayward (2000) suggest one can distinguish between 3–5 and 8–10 different values, respectively in the frequency range of 2–300 Hz and 70–1000 Hz. Birnbaum (2007) pointed out that studies provide different results on perception of vibration intensity. One study suggested that the intensity JND is 0.4–2.3 dB (Kruger, 1996). Another study suggested that one may distinguish between four different intensity levels (Gill, 2004).

It is, in other words, evident that pitch and intensity perception through the tactile channel is limited. Nevertheless, this suggests that one can indeed distinguish frequencies and intensity levels from each other. That is, they represent perceivable parameters that are analogous to musical parameters.

As for the sensing of other stimuli that can be musically related, Okazaki showed that consonant relationships between haptic and auditory stimuli can be perceived (Okazaki et al., 2013). Consonance is related to harmonic relationships. In music this is commonly found in the spectrum of complex tones as *partials* that are harmonically related to the fundamental, $f_0, 2f_0, 3f_0$... Partials are the individual frequencies in the spectrum that are relative to the fundamental frequency (f_0). Another example of harmonic relationships is between the fundamental frequencies of complex tones. For example, the relationship between f_0 and $2f_0$ is an octave. These are called either harmonic or melodic intervals depending on whether they occur simultaneously (former case) or sequentially (latter case).

Rhythm. Iterations of events occurring slower than 16–20 times per second are often perceived as rhythmic in auditory perception (Sethares, 2007). Iterations of events occurring faster than 16–20 times per second are perceived as having a pitch. While such a divide between rhythmic and pitched signals exists for audible signals, one can see from Table 2.1 on page 13 that the mechanoreceptors respond to different frequency ranges. The findings of Young et al. (2013) suggest that square waves with frequencies lower than 20 Hz are perceived as rapid clicks, a finding that is similar to that of auditory perception. Giordano and Wanderley (2013) explain that Brown et al. (2005) used small rhythmic sequences to to create *tactons* (tactile icons).

Timbre. The spectral content of a sound signal is related to what is often called *timbre*. Timbral qualities of sound are usually associated with metaphors such as "bright" and "dull". The spectrum of a sound is related to such metaphors, for example, how a sound with much energy in high frequencies can be described as "bright". The non-linear frequency perception of human hearing suggests that also amplitude may effect the perceived timbre, since the Fletcher-Munson curve flattens when the overall loudness increases. It is problematic to speak of timbre with emphasis on the spectral content as well, since it usually changes rapidly throughout the duration of the tone (Halmrast et al., 2010). However, in the thesis I will use timbre to couple metaphors to the spectral content of a sound.

With respect to tactile sensing, Picinali et al. (2012) showed that stimuli with different spectral content can indeed be differentiated. Russo et al. (2012) suggest that one may perceive differences between musical signals coming from different instruments through tactile perception. In other words, one may perceive timbral differences of musical signals through tactile perception.

Roughness. Given the explanation of the role of the spectrum of a sound in determining the perceived timbre, roughness can be seen as related to timbre. Regarding the spectral content of signals, the perceptual attribute *roughness* can be sensed by being subjected to an amplitude modulated signal (Park and Choi, 2011). The reason for pointing out this perceptual category is the fact that the degree of roughness can be used for conveying information on musical parameters.

Amplitude modulation denotes time varying modulation of the amplitude of a signal. This is commonly achieved by modulating the amplitude of one oscillator using another

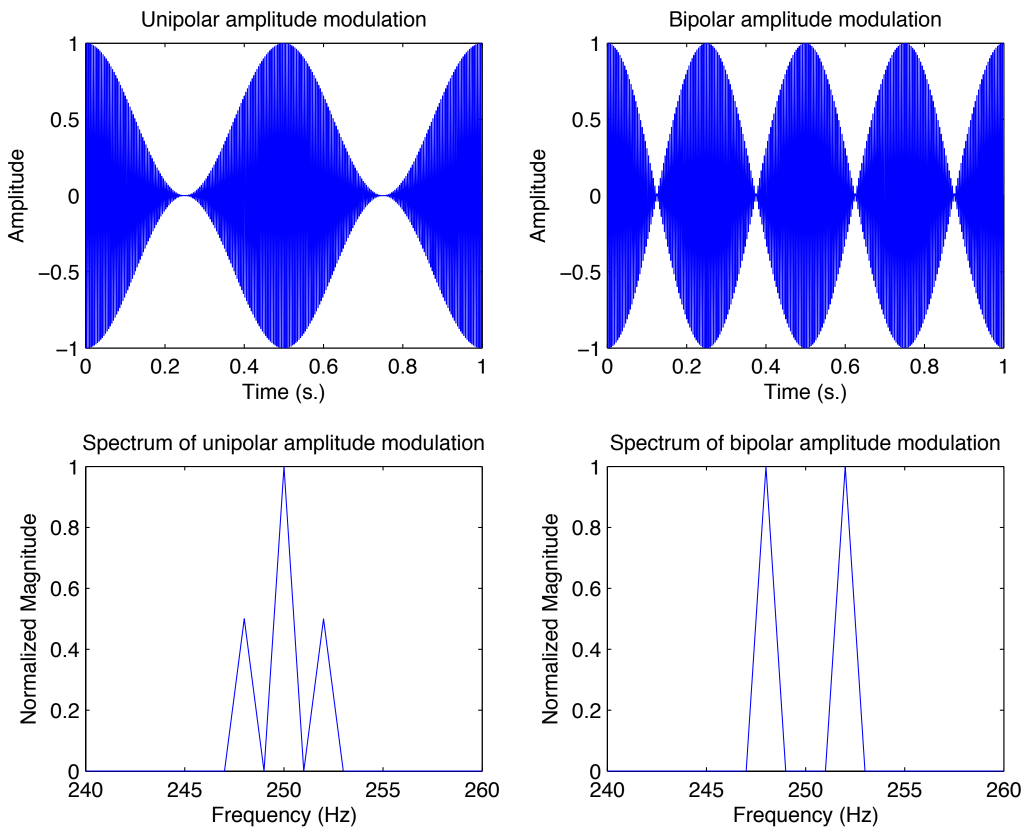


Figure 2.4 Amplitude modulation of 250 Hz sinusoidal signal by a unipolar signal (left) and bipolar signal (right) of 2 Hz. This signal creates a pulsating sensation.

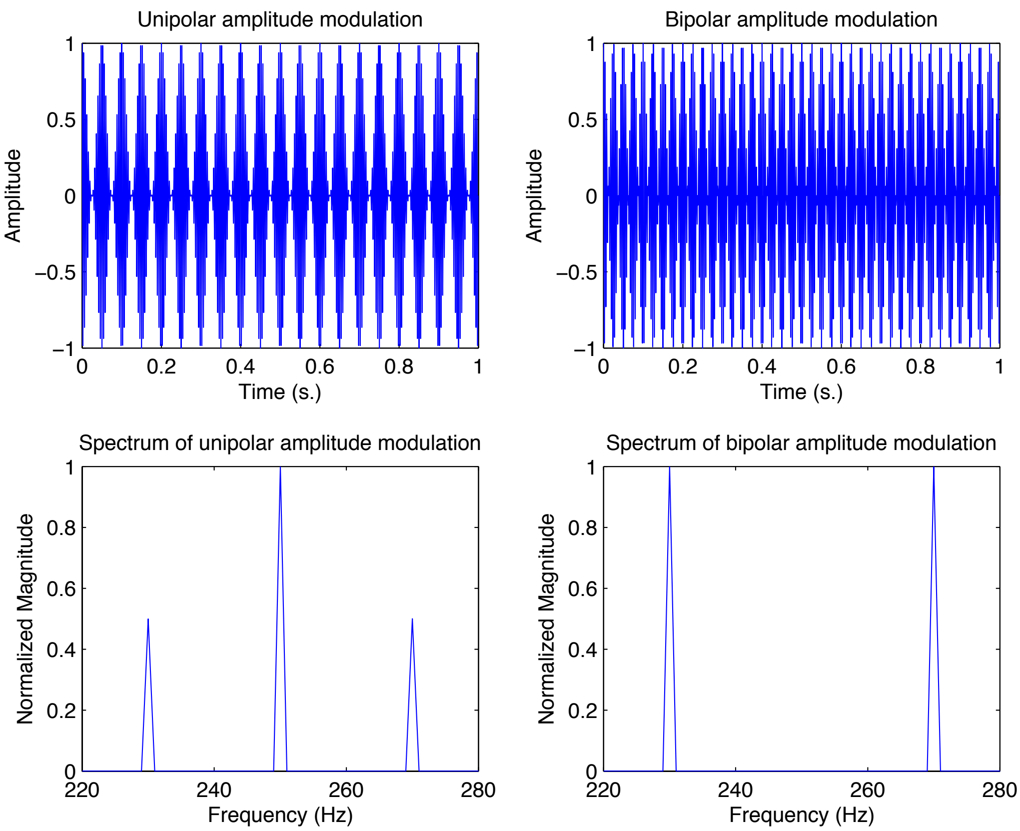


Figure 2.5 Amplitude modulation of 250 Hz sinusoidal signal by a unipolar signal (left) and bipolar signal (right) of 20 Hz. Creates a signal that is perceived a degree of roughness.

oscillator. The frequency of the modulating oscillator is called *modulation frequency* (f_m), while the frequency of the modulated oscillator is called *carrier frequency* (f_c). When using sinusoidal oscillators the spectrum of amplitude modulated signals will contain $(f_c - f_m) + (f_c + f_m)$ or $f_c + (f_c - f_m) + (f_c + f_m)$, depending on whether the modulation signal is *bipolar* (former case) or *unipolar* (latter case). A bipolar signal may have both positive and negative amplitude, while a unipolar signal may only have either positive or negative amplitude (Roads, 1996). In the case of bipolar amplitude modulation, the fundamental "disappears" from the spectrum.

Using both unipolar and bipolar amplitude modulation, having a low f_m will give a more pulsating signal, while a higher f_m will give result in a signal that can be perceived as roughness. Brown et al. (2005) suggested the degree of roughness can be used to convey information through tactile sensing. Roughness corresponds to the frequency of f_m. Figure 2.4 shows a 250 Hz sinusoidal signal modulated by a unipolar and bipolar signal both with a frequency of 2 Hz. When fed to an actuator, this signal can be perceived through tactile sensing as pulsating. Figure 2.5 shows a 250 Hz sinusoidal signal modulated by a unipolar and bipolar signal with a frequency of 20 Hz. The resulting stimuli in this case is a degree of roughness. Both figures show the signals in the time and frequency domain representation of the signals.

Giordano and Wanderley (2013) did however point to the fact that the definition of roughness is not uniformly defined in the literature. For instance, Rovan and Hayward (2000) refer to the complexity of a the spectrum as the key premise for the perception of the degree of roughness. Here a sinusoidal tone would have a lesser degree of roughness than a more complex tone such as a square wave. I will refer to roughness when the stimuli are created using amplitude modulation.

2.8.2 Spatial Domain

Vibrotactile stimuli in the spatial domain entails placing actuators on different locations of the body. By acuity, pattern recognition, and numerosity, Giordano and Wanderley (2013) refer to the ability to respectively: (1) localize stimuli accurately; (2) recognize tactile patterns; and (3) recognize multiple stimuli simultaneously. These will not be explained in more detail since they were not pursued in the implementation in Chapter 3.

Tactile Illusions. It is possible to create what is known as *tactile illusions* (Hayward, 2008). A well known tactile illusion is the "cutaneous rabbit" (Geldard and Sherrick, 1972). The principle behind this illusion is to use actuators to simulate the sensation of a rabbit jumping up the arm. This illusion can be illustrated by placing actuators along the arm. Pulses are then sent to the individual actuators sequentially to create an illusion of apparent motion along the arm.

Drawing upon this research, Miyazaki et al. (2010) experimented with vibrotactile feedback that created the sensation of the "cutaneous rabbit hopping out of the body". Lim et al. (2012) showed that vibrotactile stimuli with ascending and descending frequency provided to separate fingers, can create an illusion of apparent motion between the fingers.

Attention. Vibrotactile cues can be used to provide notifications to attract attention to certain bodyparts. Consider how insufficient pressure on a string on a guitar will cause it to buzz. As a consequence this buzz will produce vibration that attracts attention to this part of the finger. The performer therefore gets feedback on which string that is not being sufficiently pressed down.

2.8.3 Actuators

Table 2.3 provides an overview of different vibrotactile actuators. An explanation of all these different kinds of actuators, as well as algorithms for controlling haptic feedback, is beyond the scope of this thesis. Rather, the focus is directed towards two types of of actuators, namely circular vibration motors (ERM motors) and small surface mount loudspeakers (Voice coils).

From Table 2.3 one can see that these actuators have different characteristics. They are both considered to be available to the general public and they are low cost. They are not very complicated mechanically, nor do they require complex circuitry. Undoubtedly, the cost is a significant factor when wanting to create new devices. Also, since they do not need complex circuitry, one can create prototypes rapidly.

Table 2.3 Different vibrotactile actuators and their characteristics. Table based on (Choi and Kuchenbecker, 2012, p. 5)

Actuator	Affordability/ Availability	Mechanical Simplicity	Electric Simplicity	Customizability	Expressiveness
Solenoid	Medium	Medium	Medium	Medium	Medium
Generic Voice Coil	Low	Low	Medium	High	High
Vibrotactile Voice Coil	Medium	High	Medium	Low	High
ERM Motor	High	High	High	Low	Low
Piezoelectric Actuator	Low	Medium	Low	Medium	High

The voice coils score higher on "expressiveness" since they can be fed complex signals such as audio signals. This is advantageous when wanting to provide musical vibrotactile feedback. In contrast, the vibration motors cannot reproduce signals that are equally complex since only the vibration speed of the motor is the only variable parameter. An elaboration on these actuators follows in chapter 3.

2.8.4 Previous Work

The term *haptic display* is often used for devices that can convey information through haptic stimulation. Many of these attempts originate from other areas of research, such as virtual reality, where a music is not in focus (Sziebig et al., 2009; Kim et al., 2010). Table 2.4 on the facing page contains an overview of some attempts of integration of haptics in DMIs and musical applications. This is by no means an exhaustive list. Rather, it provides an overview of some attempts that I find to be relevant for this thesis. In this section I will elaborate on some of the examples.

Schumacher et al. (2013) showed how vibrotactile feedback could be used to provide feedback on the state of a live-electronics environment. Rhythmic cues were conveyed by using circular pager vibration motors to create a haptic clicktrack. Using the same vibration motors, attention could be directed to different parts of the body. Attention to the different body parts could in this case provide information on the parameters in the software. The benefit of such feedback can be to relieve the performer of reliance on visual feedback, which is quite common when performing with live-electronics environment.

Through the *Cutaneous Grooves* project, music was composed for the sense of touch (Gunther and O'Modhrain, 2003). The stimuli were provided to the audience using a full-body suit. By providing stimuli to different parts of the body, Giordano and Wanderley (2011) created a learning interface for novice guitar performers that can convey information for instance on the beat of the music.

Moss and Cunitz (2005) used a stylus interface for controlling sound and providing kinesthetic feedback in the *Haptic Theremin*. The Haptic Theremin was "fretted" using kinesthetic stimuli to aid the performer to locate pitches. Another well known example of both kinesthetic and vibrotactile feedback is the *Modular Feedback Keyboard* that was developed at ACROE (Cadoz et al., 1990). Here a physical model is the basis for the accurate and precise haptic feedback. More recent work at ACROE has resulted in the ERGOS system which can be used for a wide range of musical purposes (Cadoz et al., 2003; Sinclair, 2012).

Table 2.4 Some examples of haptic integration in DMIs.

DMI	Reference
Audio-Tactile Glove	Young et al. (2013)
Breakflute	Birnbaum (2007)
Cutaneous Grooves	Gunther and O'Modhrain (2003)
Feedback for live-electronics environment	Schumacher et al. (2013)
Feedback for physical model	Chafe (1993)
Glove display	Sachs (2005)
Guitar Learning Display	Giordano and Wanderley (2011)
Haptic Drum and Cellombo	Berdahl et al. (2008)
Haptic Theremin	Moss and Cunitz (2005)
Modular Feedback Keyboard	Cadoz et al. (1990)
Sensory Substitution Display	Egloff (2011)
Tactile Ring	Rovan and Hayward (2000)
Viblotar and Vibloslide	Marshall and Wanderley (2006)

Loudspeakers are in fact large voice coils that can also be used as actuators. The *Viblotar, Vibloslide, Haptic Drum*, and *Cellombo* used loudspeakers for sound generation and vibrotactile feedback. This means that the loudspeakers that are embedded in the DMI produces vibrotactile feedback as a byproduct of the sound production (Marshall and Wanderley, 2006; Berdahl et al., 2008).

In the *Breakflute*, vibrotactile feedback is provided to the fingertips using small voice coils. These were small enough to fit in the toneholes of a flute (Birnbaum, 2007). The vibrotactile signal is created by processing the audio signal with respect to audio features and tactile frequency perception compensation. Chafe (1993) used a voice coil actuator to to provide vibrotactile feedback using the audio signal generated by a physical model. Egloff (2011) used surface mount speakers (voice coils) to create a vibrotactile sensory substitution display.

Sachs (2005) created a mechanical motion capture controller for the hand with an inherent vibrotactile display. Here vibration motors were distributed on different locations on the hand such that a braille-inspired strategy could be used to convey musical information. Both Rovan and Hayward (2000) and Young et al. (2013) proposed designs that can provide vibrotactile feedback for DMIs with open air controllers. In the latter attempt, a glove design with actuators placed on the back of the fingers was made. Thus, the glove could provide stimuli to the individual fingers.

Rovan and Hayward (2000) pointed out the issues of heavy reliance on visual feedback, proprioception, and egolocation when using open air controllers. Visual feedback from e.g. a computer screen can be disturbing and inadequate. Reliance on proprioception and egolocation introduces inaccuracy and imprecision to the performance with open air controllers. Undoubtedly, accuracy and precision is sought after in musical performance. With this in mind, they developed the *tactile ring* which is a vibrotactile actuator that can be placed on the finger of the performer. The tactile ring can deliver a wide range of signals in the temporal domain and can be used to augment an open air controller design.

2.9 Summary

This chapter dealt with the central theory for answering the thesis problem. Several areas of research were introduced to explain the multidisciplinary nature of the thesis problem. The terms motion, action, gesture, tactile perception, and multimodal perception were explained in the scope of embodied music cognition. Here a categorization of motion, action and gesture with in respect to embodied music cognition and musical practice was provided. This categorization can also be used to describe interaction with DMIs.

The source of the thesis problem lies in the absence of a mechanically vibrating relationship between the sound generator and the controller. That is, the absence of vibrotactile feedback. It was therefore necessary to provide an overview of the theory behind DMIs, with emphasis on open air controllers. Open air controllers are as Rovan and Hayward (2000) pointed out, a special case since, e.g. since the performer may not be touching a physical surface.

Another consequence of the inherent decoupling of the sound generator and the controller is the need for mapping strategies. Theory and tools for creating mapping strategies were presented. While the mapping describes the coupling of the sensor input and the parameters of the sound generator, it is also the foundation for the action-sound relationships. Vibrotactile stimuli can be seen as feedback when they vary in relation to the human manipulation of musical parameters. Therefore, one may argue that mapping is crucial for establishing action-tactile couplings.

The physiology of tactile perception was presented with focus on perception of vibration through the hands. Compared to hearing, tactile sensing has much less bandwidth. Nevertheless, vibrotactile stimuli are distinguishable with respect to parameters such as frequency and intensity. Musical parameters in both the temporal and spatial domain that may be conveyed through vibrotactile feedback were presented. Here the emphasis was on temporal domain. Temporal musical parameters are pitch, amplitude, timbre, roughness and rhythm. These parameters are analogous to vibrotactile stimuli that can be produced using actuators such as voice coils.

Chapter 3

Implementation

The research explained in this chapter was conducted during the semester I spent in the IDMIL at McGill University as a graduate research trainee. This research contains an exploration and assessment of hardware and software that was used to define two DMI prototypes that were described in the introduction. These prototypes were augmented with vibrotactile feedback using actuators, and vibrotactile signals were synthesized.

As presented, there exist different DMI approaches that involve the use of a open air music controller. Few of the musical instruments that use open air motion to control sound have an established playing tradition. Arguably, the Theremin is the oldest instrument where open air motion is used for control of musical sound. Throughout the existence of the Theremin, performers have been playing without any form of haptic feedback. This is because the performer controls the amplitude and the pitch of the sound generator (continuously) in open air without touching a physical surface. Performers have therefore developed gestural vocabularies to deal with e.g. the of locating pitch accurately and precisely (Montague, 1991).

Due to the issues related to DMIs, such as mapping, there is no immediate answer to the way music is to be played with DMIs. Although the Theremin itself is not a DMI in its original form, a digital reimplementation of the Theremin paradigm would perhaps seem as a natural starting point when investigating vibrotactile feedback for open air musical controllers. Consider for instance the possibility of supplementing the mentioned gestural strategies with vibrotactile feedback. The issue of haptic augmentation of Theremin related instruments has already been addressed by O'Modhrain (2001). This study targeted for instance haptic influence on playing accuracy. Therefore, I chose not to focus on issues related to the Theremin. Since there is no convention for open air controlled DMIs, I chose to approach the thesis problem by constructing two DMI prototypes. The prototypes deal with different aspects of control of musical sound.

This chapter will address the issues of:

- choosing technology for an open air controller
- choosing vibrotactile actuators to be embedded in the controller design
- defining DMI prototypes and programming audio synthesis for the DMI sound generator
- creating vibrotactile feedback strategies

3.1 Defining an Open Air Controller

There was a wide range of other technologies in addition to the Theremin, that could serve as components of open air music controllers in the IDMIL: the Buchla Lightning, Leap Motion, WiiMotes, the Microsoft Kinect sensor, the Polhemus Liberty,[1] the Radio Baton, and the Vicon V460 infrared marker based MoCap system with six M2 cameras. The Vicon system is a passive IR marker based MoCap system. That is, infrared light is projected on scene on which the subject with the attached markers is located. The cameras that are placed around the subject can pickup up the reflected light from the markers. When the marker layout is fixed on the subject in either dynamic and rigid formations, models can be defined in the computer software. This means the motion of a subject's body and individual limbs can be captured, recorded and analyzed. The marker position data can also be used to control sound synthesis. Some examples of realtime control of sound using the Vicon motion capture systems are shown in (Bevilacqua et al., 2002; Dobrian and Bevilacqua, 2003; Martin, 2011; Mamedes et al., 2013).

The Vicon system was chosen for several reasons. Although the Kinect is the least obtrusive of the mentioned devices, since it does not require the user to wear markers, the Vicon system offers higher precision, accuracy and a larger field of view. Since I chose to focus on hand motion to control sound, a full-body marker configuration was not needed. In this case the Vicon was not considered as obtrusive.

Since hand motion was targeted, the Leap Motion was a possible candidate as an open air controller. However, since the it focuses on motion within a small field of view, use of the Leap Motion was dismissed. I wanted to allow the performer to execute motion with large magnitudes and within a larger space.

[1] http://www.polhemus.com/?page=Motion_LIBERTY

The Vicon system was already in place, mounted alongside a circular eight-speaker setup in the IDMIL. The system was already set up with access to realtime export of the sensor data using QVicon2OSC. This was seen as advantageous. Having access to the Spat externals for Max MSP and the eight-speaker setup allowed exploration with spatialization of sound. This was initially seen as an interesting feature to be included in the DMI design.

3.2 Choice of Actuators

To find a suitable actuator for the open air controller, two types of actuators were assessed. These were flat vibration pager motors and voice coil actuators. Although many different actuators exist the selected actuators were chosen due to the low cost and the availability, but also because they perform adequately with respect to the vibrotactile stimuli they can provide. Here the rationale for the choice of actuators is presented.

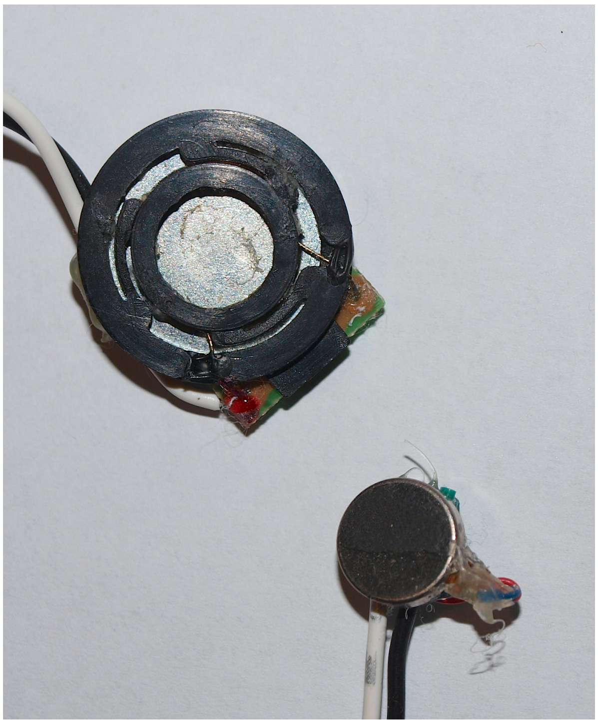


Figure 3.1 Picture shows the 11 mm surface mount speaker (voice coil) on the top and the 8 mm circular vibration motor on the bottom.

3.2.1 Vibration Motors

Vibration motors are often powered by a DC signal. The produce vibration by spinning an unbalanced mass. This means that the once the mass starts moving, vibration is produced. By varying the voltage of the electrical signal supplied to the vibration motors, the vibration speed can be controlled. One method for doing so implies using a pulse width modulation (PWM) signal. This can be achieved using an Arduino[2] micro controller. The PWM signal resembles a square wave where the frequency is fixed, and the *duty cycle* is varied. The duty cycle denotes the portion of the period of the signal where the magnitude is at the maximum. This technique can be used to control the effective voltage supplied to the vibration motor. E.g., a full duty cycle would yield a maximum effective voltage.

Vibration motors are often found in mobile phones and game controllers. In this way they can provide notifications or feedback on the interaction with the system, for instance when the a phone is in silent mode. Although vibration intensity (i.e. motor speed) is the only variable parameter as (Table 2.3), the can still convey musical information. The types of feedback signals one may create are for instance gradually varying intensity. They can also be laid out spatially which makes it possible to create stimuli in the spatial domain. Such vibrotactile stimuli were exemplified in Chapter 2 with reference to e.g. Schumacher et al. (2013).

With respect to the explained possible vibrotactile focal areas, four circular 8 mm vibration motors (Figure 3.1) were placed inside a glove. The actuators were placed on the thumb, the index finger, the little finger, and inside the palm. Using an Arduino Nano v3 micro controller and Puredata, I experimented with creating discernible vibration stimuli. The Arduino was controlled using the `[arduino]` object in Pd. The Arduino then used the PWM outputs to control a driver circuit. In turn, the driver circuit controlled the effective voltage supplied to the motors. By increasing the vibration intensity in the one motor while decreasing it in the other, the hope was to achieve a similar effect as found in Lim et al. (2012), that is, apparent motion between the fingers. Notification stimuli were also explored as well as sequential notification patterns, e.g. circular patterns.

For more flexibility, this experimental setup was ported to another setup consisting of libmapper, SuperCollider and Firmapper. The control of the PWM signals was performed in SuperCollider. Libmapper device inputs were created so that one could connect an input device to the algorithms controlling the vibration patterns. With the

[2] www.arduino.cc

libmapper bindings for SuperCollider, the control signals were sent to the *Firmapper*[3] program. Firmapper is a libmapper extension for the *Firmata*[4] protocol for Arduinos. Firmata allows on-the-fly reconfiguration of the inputs and outputs of an Arduino by using e.g. Pd. Firmapper appears automatically in the libmapper GUI. By using the Firmapper GUI one can therfore configure the inputs and outputs of an Arduino while also creating and destroying libmapper inputs and outputs. The configuration can be saved to a file or on in the memory of the Arduino.

3.2.2 Voice Coil Actuators

Voice coil actuators come in different shapes and can be controlled using an audio signal, although they usually require amplification. Egloff (2011) used voice coil actuators to create a vibrotactile display for sensory substitution. Similar actuators were used in this thesis research for early exploration, namely 13 mm 8 Ohm voice coil actuators[5] and 11 mm 32 Ohm actuators.[6] These actuators are in small loudspeakers with a circular plastic membrane (with a hole in it) that can be glued to a surface. Primarily, they are used to as surface mount speakers. Because of this attribute, they are suitable as vibrotactile actuators for the fingers since the tip of the finger can be laid on top of this hole. Another advantage is the low cost of the actuators, they only cost around 3$ each. Since the chosen voice coil actuators are in fact small loudspeakers, they can be fed an audio signal. This means that they can produce complex signals.

3.2.3 Final Selection of Actuators

The assessment of the vibration motors entailed stimuli belonging to the spatial domain. Primarily, tactile illusions and attention cues were investigated, but also rhythmic which belongs to the category of the temporal domain. While the explored stimuli were interesting themselves, the possibilities of creating stimuli that could convey information belonging to the temporal domain such as pitch, amplitude, rhythm, roughness, and timbre with the same resolution as the voice coils was found to be less probable. That is, voice coils can also be used for creating stimuli in the spatial domain, but also offer higher resolution with respect to stimuli in the temporal domain. I therefore chose to dismiss further use of vibration motors, since I wanted to focus on vibrotactile feedback strategies belonging to the temporal domain. As explained vibrotactile feedback

[3] https://github.com/RDju/firmata-mapper/pull/2
[4] http://firmata.org/wiki/Main_Page
[5] http://www.parts-express.com/pe/showdetl.cfm?partnumber=297-214
[6] http://www.parts-express.com/pe/showdetl.cfm?partnumber=297-228

belonging to the temporal domain involves vibration stimuli that are closely related to musical signals.

3.3 Hardware measurements

Although voice coil actuators can be connected straight to the mini jack output of a laptop, the signal can sometimes become too weak for tactile stimulation. Therefore, a Sparkfun class D mono audio amplifier[7] was chosen for signal amplification. This amplifier can run on 2.5–5.5 V DC and 1.4 W maximum with an 8 ohm load, also it is a low cost device (around 8$).

The different components of a signal chain may alter the signal properties. Any of these components may be considered as filters. While there are many ways in which a signal may be effected by filters, the alteration of the frequency content is an important one with respect to musical signals. It is so because the filtered signal would sound different if the frequency content is altered. Similarly, the vibrotactile signals may be perceived differently as well if the frequency content is altered. By obtaining the frequency response of the components in the signal chain one can therefore study their influence of the on the signal. If needed, filtering can be applied as a countermeasure.

The frequency response $H(f)$ of a device can tell the manner in which certain frequencies are amplified or attenuated, as well as whether the phase of these frequencies are shifted (Moore, 1990). The frequency response of a system can be obtained by taking the discrete Fourier transform of the impulse response $h[n]$ of the measured system,

$$H\left(f\right) = \sum_{n=0}^{N-1} h\left[n\right] e^{-2\pi i f n/N}. \tag{3.1}$$

In collaboration with Marcello Giordano in the IDMIL, the frequency response of the Sparkfun amplifier was measured. This was performed by sending synthesized exponential sine sweeps through the amplifier and recording the output (Farina, 2000). A sine sweep is a sinusoidal signal start starts at one frequency and sweeps exponentially to another. The idea is that the recorded response will tell us how the system, to which the sinesweep is used as input, modifies the phase and amplitude of the frequencies within the sine sweep. Sinesweeps were generated based on Berdahl's Matlab script.[8] Matlab was also used for analysis of the frequency response.

Using a DC bench power supply set to 3V and a sinesweep from 1–2500 Hz, a

[7] https://www.sparkfun.com/products/11044
[8] https://ccrma.stanford.edu/realsimple/imp_meas/Sine_Sweep_Measurement_Theory.html

relatively flat frequency response was obtained of the amplifier. The chosen voltage of the bench supply was due to the fact that two AA batteries were going to be used when powering the amplifier in the setup. A National Instruments acquisition board (USB-4431) was used to record the sinesweep response.

If $x[n]$ is the sinesweep signal and $y[n]$ is the output of the amplifier they relate to impulse response of the Sparkfun amplifier $h[n]$ in the following way: $x[n] * h[n] = y[n]$. Here $*$ denotes convolution. Since convolution in the time domain is the same as multiplication in the frequency domain,

$$H(f) = \frac{Y(f)}{X(f)}. \tag{3.2}$$

With respect to decibels (dB), the magnitude response of the Sparkfun amplifier is,

$$20 \log_{10} |H(f)| . \tag{3.3}$$

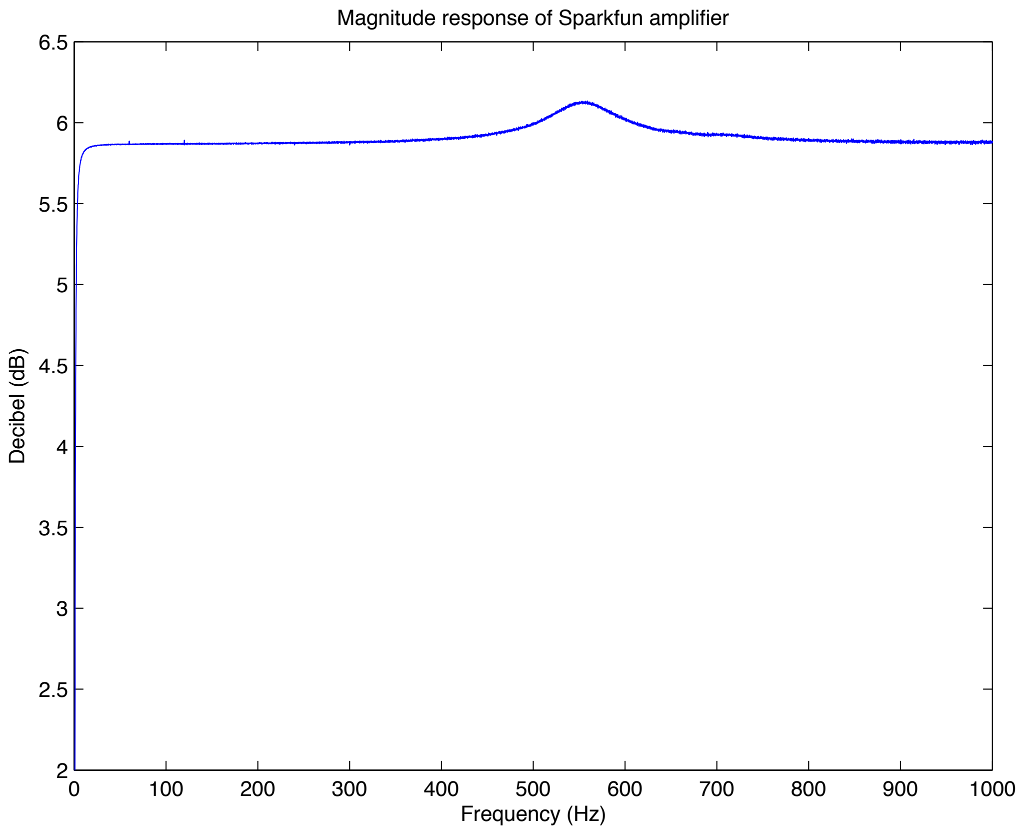


Figure 3.2 Frequency response of the Sparkfun class d mono audio amplifier.

As seen in Figure 3.2 on the previous page, the frequency response of the Sparkfun amplifier can be considered acceptable. Therefore, no filtering was considered to be necessary. Although it would have been interesting to know the frequency response of the actuators to see if they have a flat frequency response as well, this is a more complicated matter. While it is possible to measure the displacement of the membrane using a single axis accelerometer, this proved to be harder than measuring the response of the amplifier. The accelerometer (PCB Piezotronics 352C22) that was used in the attempt has to be mounted to the surface of the membrane surface using wax. However, both the 13 mm and the 11 mm actuators are quite small and it proved to be difficult to mount the accelerometer to the membrane such that it did not fall off during the sinesweep. The idea of capturing this response was abandoned due to time constraints.

3.4 Wireless vs. wired implementation

A wireless approach was initially thought of as less obtrusive. The thought of using an Android based telephone for tactile signal generation was explored in collaboration with Marcello Giordano in the IDMIL. Here *libpd* was used for tactile signal synthesis (Brinkmann et al., 2011; Brinkmann, 2012). The libpd library can be used in conjunction with several programming languages, such as C/C++, Java, Objective-C etc. This allows the programmer to embed Pd patches in the given environment. That is, libpd provides the opportunity to use Pd as a synthesis engine on different platforms. By sending OSC messages from a device to the platform running libpd, the signal synthesis can be controlled.

The battery on the phone itself can also power the Sparkfun amplifier by soldering wires onto the micro SD card (Figure 3.3). 3.3 V was then supplied to the Sparkfun amplifier while the libpd-based application was running on the phone. Although this approach worked (Figure 3.7) there was considerable latency on the signal synthesis. In certain cases it might still function properly, e.g. if the tactile cues are changing relatively gradually. However, from a general point of view it was found to be intolerably high.

To solve the issue of latency there might be several solutions: (1) synthesizing signals with a lower sampling rate; (2) using a more powerful Android based phone; (3) using a device less prone to audio latency, such as an iOS device with a low level synthesis library (Bryan et al., 2010); or (4) simply transmitting the audio signal itself wirelessly. These possible solutions were not explored due to time constraints. Although a wired approach makes the glove setup more obtrusive it does allow synthesis of signals with

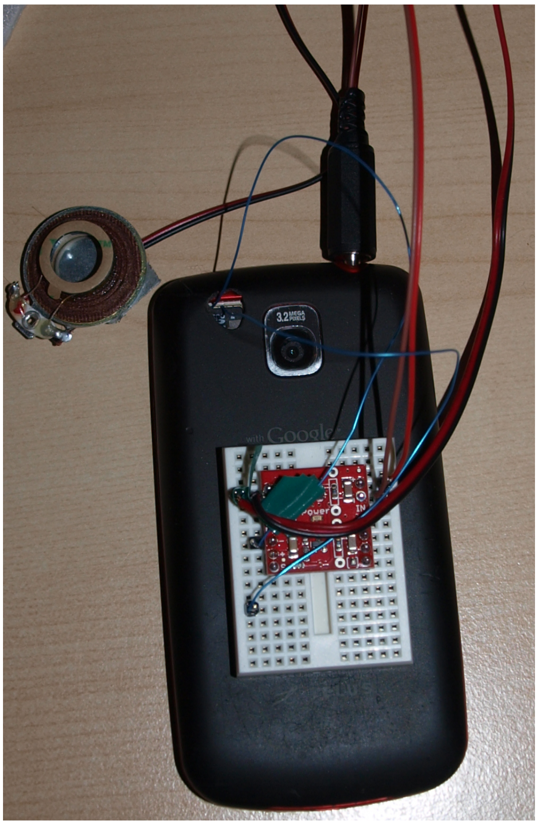


Figure 3.3 Wireless attempt with Android phone (LG optimus 1 p500h) as vibrotactile signal generator. The Sparkfun amplifier is placed in the breadboard on the back of the Phone. A 13 mm voice coil actuator is shown to upper left in the picture.

significantly lower latency. I.e., latency was seen as a more crucial issue than the obtrusiveness introduced by the wires.

3.5 Exploration and Assessment of Hardware and Software

The Vicon frame rate was set to 100 Hz. A fixed marker setup of the right hand was defined both physically with marker placement, and within the Vicon IQ software. This is necessary for the QVicon2OSC to be able to send marker position data. Another advantage is that the tracking might be more robust. For the exploration of the setup, a virtual space was defined within the circular camera setup in the IDMIL, 4m x 3m x 2.5m approximately. The Vicon MoCap data was acquired on one computer. The position data of the marker located on the middle finger was sent in real time via OSC using QVicon2OSC[9] to an iMac running Max MSP. I.e., this was the only marker taken into account when capturing the motion of the hand. The iMac also had the *Spat* externals for Max MSP installed in addition to being connected an eight speaker setup via a MOTU sound card. Spat is a sound spatialization software developed at IRCAM.[10]

Since a tradition for DMIs using open air controllers does not exist, a test prototype had to be defined such that potential vibrotactile feedback strategies could be explored. Surrounding the performer in a circular configuration around the virtual space were eight loudspeakers. Using Spat it is possible to mimic an acoustic field in which sound sources can be placed with respect to spatial orientation. Four different points in 3D space were defined. To each point a drumbeat was assigned.

Using the coordinates of the marker placed on the middle finger, distances d_i from the hand h to the drumloops i were calculated continuously using the Pythagorean theorem,

$$d_i = \sqrt{(x_i - x_h)^2 + (y_i - y_h)^2 + (z_i - z_h)^2}. \tag{3.4}$$

The thought was to locate the points to which the drum loops were assigned. The idea was use the drumloop signals with varying intensity as cues for proximity of the virtual points in the 3D. Using an iPod with TouchOSC the performer could send a message to the Max MSP patch so that the perceived center position could be recorded (Figure 3.6). The recorded coordinates could then be compared against the actual coordinates of the points. The vibrotactile signal had the largest amplitude when the

[9] http://sonenvir.at/downloads/qvicon2osc/
[10] http://www.fluxhome.com/products/plug_ins/ircam_spat

Figure 3.4 Vicon and speaker setup in the IDMIL.

hand marker was placed directly on the point.

The vibrotactile signal was created using a processed version of the drumloop audio signal. In this approach a modified version of Birnbaum's FA-SA Max MSP patch was used for processing the vibrotactile signal (Birnbaum, 2007). In my implementation in the exploration patch, the modified FA-SA patch flattens the dynamic range of the signal, filters to compensate for the nonlinear frequency response of the fingers, and filters out frequencies outside the tactile range (1000 Hz).

The vibrotactile signal was faded in corresponding to the proximity when the hand's position was closing in on the point associated with the given drumloop. Bevilacqua et al. (2002) used the Vicon motion capture system to obtain motion for triggering sound. Similarly, I chose to use rapid hand motion to trigger sound. This was implemented by taking the first difference of the marker position on the x- and y-axis. Rapid hand motion can in this case be considered to be sound producing actions.

The drum loops were panned using Spat according to the position of the virtual points in the virtual space. The drum loops kept playing as long as the hand was within a defined maximum distance to the point. Upon triggering the drum loop the tactile signal was doubled in amplitude. Thus, this DMI concept provides vibrotactile feedback on the proximity of the hand to the virtual objects as well as when the drumloop is triggered.

Localization of points was not considered to be an engaging musical task. The performer had no control over the drumloop once it had been triggered. The the size of the space made it difficult to locate the virtual points to which the drumloops were assigned. This became evident when fellow students tried the setup. Also, the edges of the virtual space had less coverage of the Vicon camera setup, which resulted in poor tracking of the hand. It was clear that the marker configuration for the hand had to be standardized if wanting to perform tests with multiple subjects. The individual differences between potential subjects can be compensated for in the Vicon software, however, this involves a calibration process. It was decided to overcome this issue by attaching markers to a glove. Although sound spatialization was considered to be an interesting feature, it may direct the attention away from the vibrotactile feedback. Further use of Spat was therefore abandoned.

3.6 Final Setup: Prototype 1 and 2

The following next steps towards defining DMI prototypes to evaluate vibrotactile feedback were also described by Knutzen et al. (2013). Two different DMI prototypes using

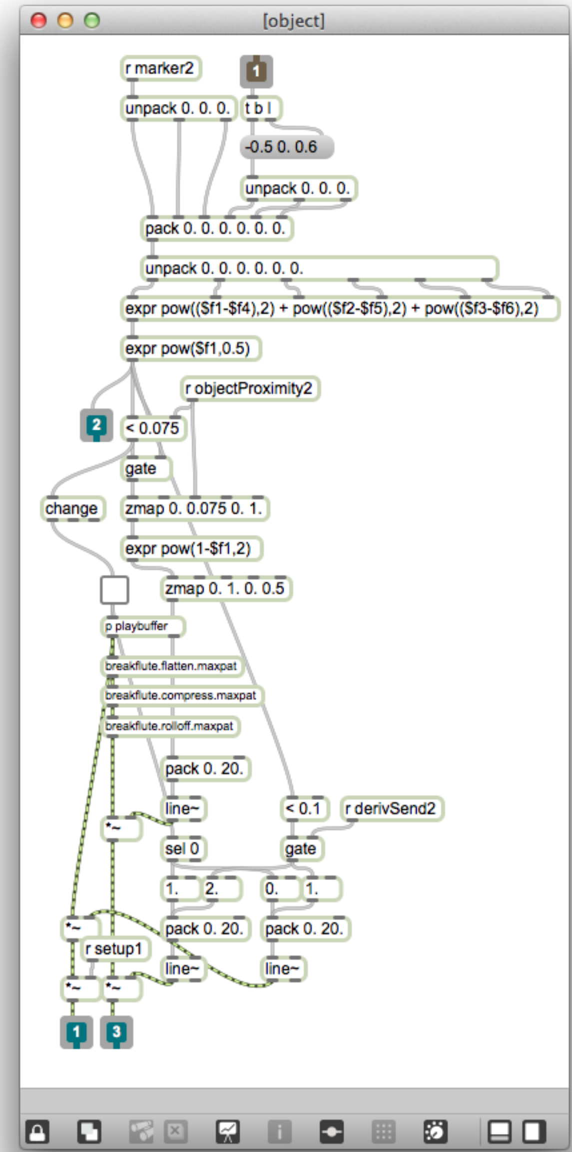


Figure 3.5 Screenshot of a Max MSP object used in the exploration. Here distance to a virtual point is calculated. The objects with the breakflute prefix are taken from Birnbaum's FA-SA Max patch. Outlet 1 is used for the audio signal and outlet 3 is used for the vibrotactile signal.

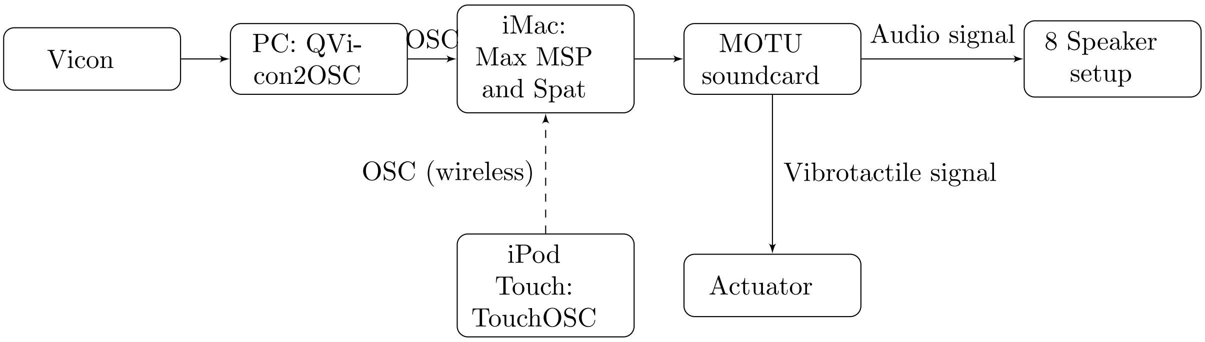


Figure 3.6 Chart of exploration setup.

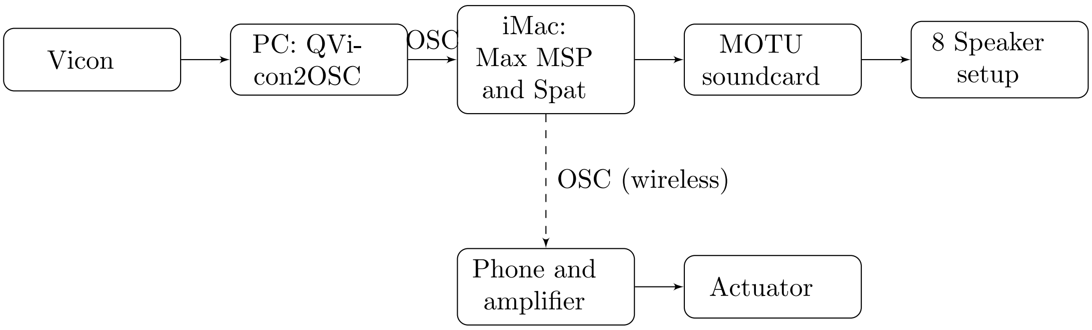


Figure 3.7 Hypothetical chart of initial setup with the Android phone as vibrotactile signal generator.

the Vicon MoCap system were created. The prototypes illustrates possible ways of controlling musical sound with open air motion. An elaboration of this work follows in this section. Based on the assessment explained above, a new configuration was set up (Figure 3.9). In this setup the marker position data of the middle finger was sent from the computer running the Vicon software to a Macbook Pro 9.2 via an ethernet cable using QVicon2OSC. Instead of using the multi-speaker setup, a headset was chosen as the main sound source, in order for the participants to hear the nuances of the sound clearly.

A fixed marker configuration was attached to a right handed utility glove (Figure 3.8). The actuators and the wires were fitted on the inside of the glove. The actuators were attached so that they would stimulate the fingertips of the index, middle, ring, and little finger of the performer. The previous attempt involved one actuator. The same musical vibrotactile signal could now be provided to all the four fingers. Thus, the vibrotactile signals became more pronounced. Another reason for choosing such a glove was for it to fit different subjects.

SuperCollider was used for synthesis of both audio and vibrotactile signals. The signal synthesis was defined using `SynthDef` class which lets you store synthesis algorithms

Figure 3.8 The vibrotactile glove with index finger actuator exposed. On the right, a closeup of the index finger actuator.

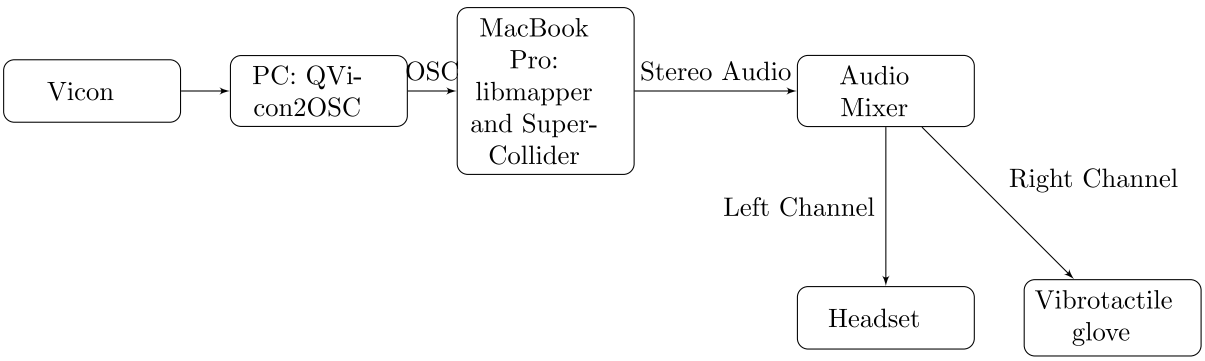


Figure 3.9 Chart of final setup.

that can be called later. The audio signal was routed to the left channel of the built-in stereo output of the MacBook Pro, while the vibrotactile signal was routed to the right channel. The two signals could then be routed to the headset and the actuators separately. This was achieved using an outboard analog audio mixer. The vibrotactile signals were sent to the Sparkfun amplifier using a minijack (Figure 3.12).

Libmapper was used to create mapping strategies for both the control of sound and vibrotactile signal synthesis. Using the libmapper GUI, the different mappings between the position data and the audio and vibrotactile synths were established. This approach also allows switching between different mappings. Two DMI prototypes were defined, each of them focusing on different aspects of controlling musical sound.

3.7 Prototype 1

Prototype 1 was a *skill*-based DMI, i.e. a DMI that focused on controlling low level parameters of the sound synthesis. In this case I chose to focus on control of pitch, timbre, and amplitude. These are parameters one can control in most traditional musical instruments. To address such a problem, I decided to create a DMI prototype that lets you select a discrete pitch belonging to a C minor pentatonic scale and triggering it with varying amplitude and timbre. The SuperCollider code for the chosen sound synthesis can be seen in Code 3.1. This code presents a modified version of a phase modulation based synth created by Cottle (2011).

```
SynthDef(\PMCrotaleMod, {
    |midi = 60, varpar = 1, art = 1,amp = 0.9,chan = 0,tactamp = 0|
    var env, out, mod, freq,tactout;
    freq = midi.midicps;
    env = Env.perc(0, art);
        mod = 5 + (1 / IRand(2, 6));
    out = PMOsc.ar(freq, mod*freq,
        pmindex: EnvGen.kr(env, timeScale: art, levelScale: varpar),
            mul: EnvGen.kr(env, timeScale: art, levelScale: 0.5));
    out = out * EnvGen.kr(env, timeScale: 1.3 * art,
        levelScale: Rand(0.1, 0.5), doneAction: 2);
        tactout = LPF.ar(SOS.ar(SOS.ar(
        out,0.874225, -1.711427, 0.838289, 1.711427, -0.712514),
                0.980631, -1.922495, 0.941894, 1.922495, -0.922526),800);
        Out.ar(chan, Pan2.ar(out*abs(amp),-1,1) + Pan2.ar(tactout,1,tactamp))
            ;
}).add;
```

Code 3.1 Sound generating algorithm for Prototype 1.

Code 3.2 shows how a libmapper input is created in SuperCollider. Each time a value is received with this input a function is called. This function defines discrete selection of notes of the C minor pentatonic scale (MIDI notes 60–77). Here a note "grid" is defined by dividing the marker position by a scalar, rounding to the nearest integer and then taking modulus 8 of the resulting value. The grid spread can be varied by selecting a different scalar for the ~`gridDivisor` variable.

```
~gridDivisor = 300; // defines grid spread
~prev = 0;
~gridInput = ~SC.addInput(
   '/gridInput', 1, $f, 'pos', -2000, 2000,{
    |signame, instanceid, value|
    var func,modvalue;
       modvalue = (value/~gridDivisor).round(1).mod(8);
       if((modvalue - ~prev) != 0, {case
      {modvalue == 0}{~gridFreq = 60; Synth(\gridSynth,[\amp,1]);}
      {modvalue == 1}{~gridFreq = 63; Synth(\gridSynth,[\amp,1]);}
      {modvalue == 2}{~gridFreq = 65; Synth(\gridSynth,[\amp,1]);}
      {modvalue == 3}{~gridFreq = 67; Synth(\gridSynth,[\amp,1]);}
      {modvalue == 4}{~gridFreq = 70; Synth(\gridSynth,[\amp,1]);}
      {modvalue == 5}{~gridFreq = 72; Synth(\gridSynth,[\amp,1]);}
      {modvalue == 6}{~gridFreq = 75; Synth(\gridSynth,[\amp,1]);}
      {modvalue == 7}{~gridFreq = 77; Synth(\gridSynth,[\amp,1]);}
      },{});
      ~prev = modvalue;
});
```

Code 3.2 Shows how a libmapper input is created in SuperColldier, as well as how the discrete note grid is defined.

```

SynthDef(\gridSynth,{
      |amp = 1,chan = 1|
      Out.ar(
            chan,RLPF.ar(
                  Impulse.ar(1)*EnvGen.ar(
                        Env.new([1,0],[0.1,0.1]),
            doneAction:2,levelScale:50)*amp,250));
}).add;
```

Code 3.3 Vibrotactile signal synthesis for the providing notifications on note selection.

3.7.1 Motion→sound

The motion to sound mapping is illustrated in Figure 3.10. Here, the x-axis of the Vicon coordinate system was divided into a discrete note "grid". Assigned to the grid were the pitches of the pentatonic scale, C4–F5. When crossing from positive to negative on the y-axis the note was triggered (sound producing action). To be able to control the amplitude, the first difference (velocity) of the marker position on the y-axis was used. Thus, depending on how fast the performer would cross from the positive to the negative side on the y-axis, the amplitude of the sound would scaled accordingly. Depending on the height of the hand, that is the marker position on the z-axis, the timbre could be controlled, low–high to "dull"–"bright" (sound modifying action). In this prototype there was no dynamic control of the control parameters once the synth is triggered.

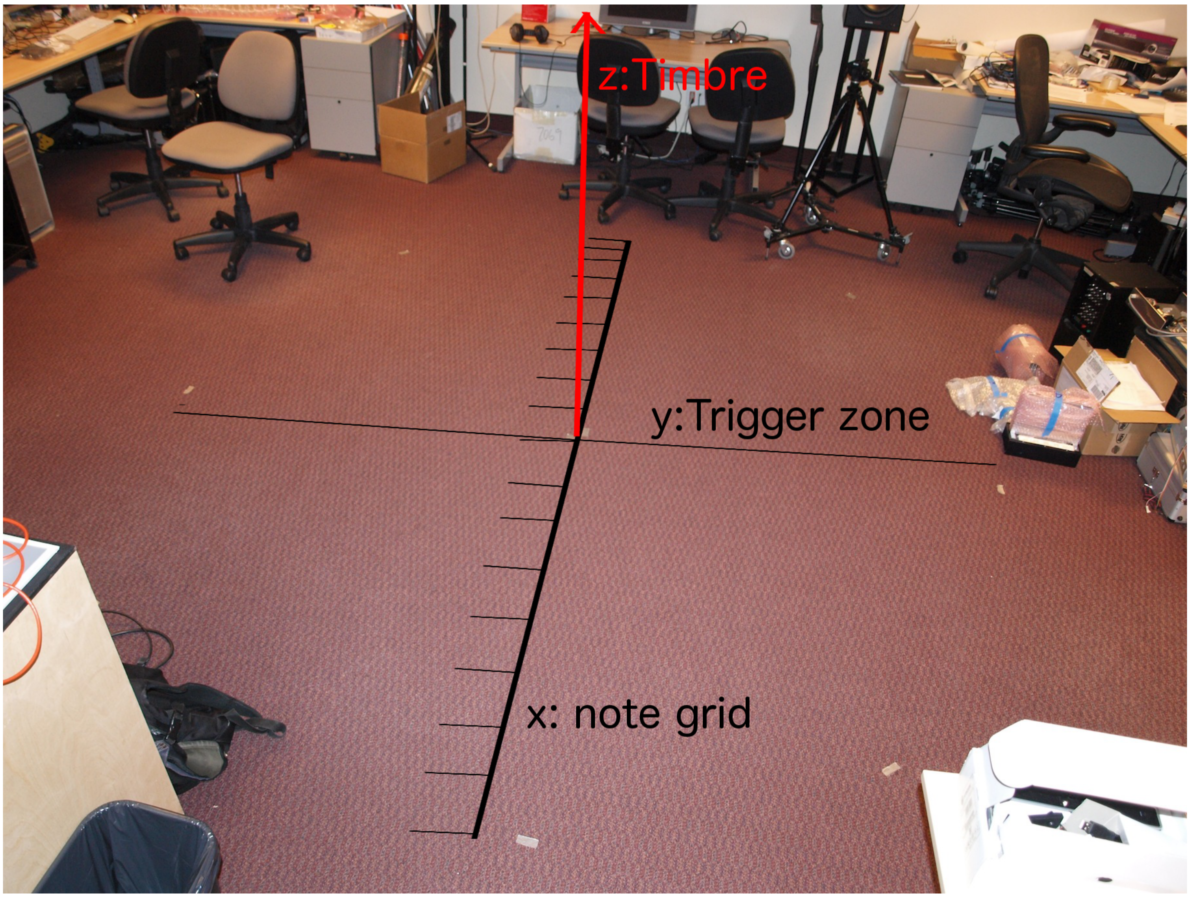


Figure 3.10 Mapping of motion to parameters of the sound synthesis in Prototype 1. The note grid is aligned with the x-axis of the Vicon coordinate system. The "trigger zone" denotes the space where notes can be triggered, the red arrow shows the z-axis that is mapped to the spectral parameters of the audio synthesis.

3.7.2 Motion→vibrotactile feedback

Not only was the vibrotactile feedback here a way of conveying information on musical parameters, but also on the actions of the performer.

- Note selection: Response to sound modifying action.
- Note triggering: Response to sound producing action.

- Amplitude: Response to sound modifying action.
- Timbre: Response to sound modifying action.

Inside the function that defines the note grid, a synth was triggered each time a new note was selected. This synth creates a vibrotactile signal for note selection feedback. It is called `gridSynth` in Code 3.3. The vibrotactile signal was produced by sending an impulse to a resonant low pass filter with the center frequency set to 250 Hz. The note selection feedback strategy was inspired by the Haptic Theremin that was created by Moss and Cunitz (2005). Here kinesthetic feedback was used to help the performer select pitches in a Theremin-like DMI setup.

Four separate vibrotactile strategies were defined by synthesizing signals in SuperCollider. All of the sound signals with varying frequency content created by these vibrotactile synths were filtered by two cascaded biquad filters with respect to the tactile frequency compensation filtering that was proposed by Birnbaum (2007). The coefficients were obtained from Birnbaum's FA-SA Max MSP patch. Four separate vibrotactile strategies that gave feedback on the mentioned parameters were were defined. The note selection feedback was used in conjunction with all of these strategies. In the following, I will provide a description of the vibrotactile feedback strategies along with exemplifications of possible outcomes of interaction with Prototype 1.

Feedback Strategy 1: Sine

This strategy involves producing sinusoidal vibration. As explained, Okazaki et al. (2013) suggested that one may perceive harmonic relationships between vibrotactile stimuli and perceived sound. To establish a harmonic relationship with the sound signal while keeping the signal within the most sensitive range of perceivable tactile frequencies, the fundamental of the sinusoidal vibrotactile signal was set to half the frequency of the sound signal ($f_0/2$). The amplitude varied in accordance with the amplitude of the audio signal. The time varying amplitude envelope of this synth was similar to the envelope of the audio signal. Thus, in this feedback strategy there was no feedback on the timbre of the played sound. A rapid sound producing action on the y-axis will therefore resulted in a sinusoidal signal with a high maximum amplitude. Conversely, a slow sound producing action would trigger a vibrotactile signal with a low maximum amplitude.

Feedback Strategy 2: Bursts

With this strategy the created vibrotactile signals did not share similarities with the audio signal. Here a burst of impulses was created once the sound was triggered. The vibrotactile signal was made by filtering the impulses with a resonant low pass filter where the filter frequency was equal to the fundamental (f_0) of the sound signal. The impulse iteration speed was controlled by a linearly decaying envelope, meaning that the impulse iteration frequency decreased with time. The maximum amplitude of the burst was the same as for the audio signal, which means that amplitude feedback was provided. The burst length was controlled by the height of the hand. Thus, feedback on the timbre of the sound was provided.

A rapid sound producing action with the hand held high would then trigger a long burst with a high maximum amplitude. Conversely, a slow sound producing action with the hand held low would trigger a short burst with a low maximum amplitude. With this strategy the note selection feedback was provided by short sinusoidal pulses. The reason for using a different note selection feedback in this strategy was because the burst were very similar to the pulses provided by the note selection feedback. Thus, it would perhaps become confusing for the performer to distinguish between note selection feedback and note triggering feedback.

Feedback Strategy 3: Amplitude modulation

Here the vibrotactile signal was made using unipolar amplitude modulation. The carrier frequency was set to $f_0/2$ of the sound signal. The time varying amplitude envelope was similar to the sound signal, and the maximum amplitude was varied in accordance with to the maximum amplitude of the sound signal. The modulation frequency of the amplitude modulation was varied with the height of the hand, low–high to 7–20 Hz, meaning, that the vibrotactile signal was felt as pulsating when the timbre is "dull", while roughness was felt when the timbre is "bright". The behavior of the vibrotactile signal therefore varied with respect to amplitude upon triggering as with the first strategy.

Feedback Strategy 4: Filtered audio signal

The vibrotactile signal in this strategy was the audio signal that had been filtered with respect to the vibrotactile frequency perception equalization as found in Birnbaum's FA-SA application (Birnbaum, 2007). The vibrotactile signal was also low pass filtered at with a cutoff frequency at 800 Hz to roll of some audible frequencies. Since the "edge" of the vibrotactile frequency perception range was at 1000 Hz, filtering at 800 Hz would not impair the vibrotactile signal noticeably. This vibrotactile signal would therefore provide feedback on the f_0 of the signal, the maximum amplitude and the timbre.

3.7.3 Example of Interaction

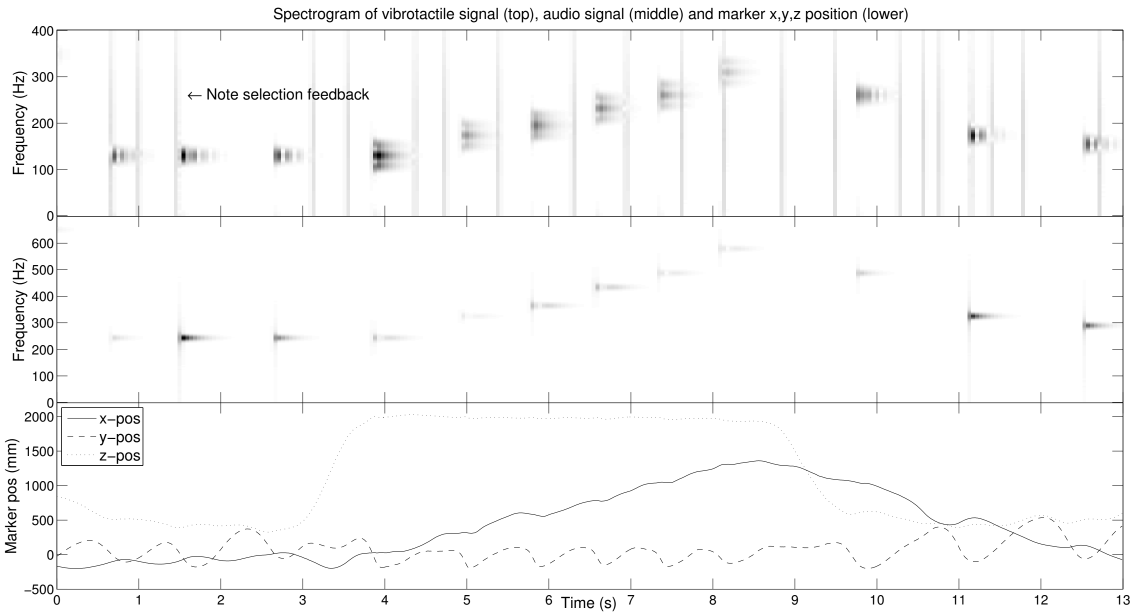


Figure 3.11 The plot shows spectrograms of the vibrotactile signal made with feedback strategy 3, as well as the audio signal. Aligned in time with both of the spectrograms is the marker position with respect to the x-, y- and z-axes.

In Figure 3.11, interaction with corresponding audio and vibrotactile feedback is exemplified (Knutzen et al., 2013). The spectrograms were created using the MIRtoolbox (Lartillot et al., 2008). Spectrograms show the energy of the different frequencies in a signal over time. The spectrogram in the middle shows the audio signal while the top spectrogram shows the vibrotactile feedback signal of feedback strategy 3 . The bottom plot shows the marker position with respect to the x-, y- and z-axis, the units are in millimeters. Thus, sound producing and sound modifying actions are represented with the marker position data, aligned with a visual representation of the resulting audio and vibrotactile signal.

Pointed out in the upper spectrogram is one of the note selection feedback pulses. The note selection feedback can be seen as grey vertical lines. In addition to the note selection feedback, one can see feedback strategy 3 represented in the spectrogram. At time 4 seconds one can see that the modulation frequency of the amplitude modulated vibrotactile signal is relatively high compared to the preceding note judging from the spread in the spectrum. Thus, the vibrotactile signal is perceived as having a high degree of roughness. At the same time one can see how the z-position has a high value, meaning that the hand is held high. This means that the triggered note is "bright". From time 5–9 seconds one can see a staircase like curve on the x-position. This motion indicates stepwise note selection.

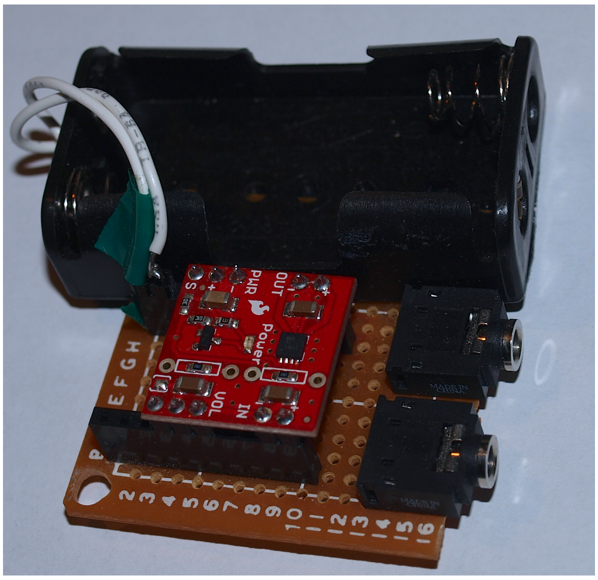


Figure 3.12 Sparkfun amplifier with mini jack plugs and battery holder.

3.8 Prototype 2

Prototype 2 was a *model*-based DMI, i.e. it focuses on higher level control of musical parameters. Here the sound was generated by playing back a drumloop stored in a buffer (Code 3.4). The signal was then filtered by a resonant low pass filter (`RLPF` is the code). The tempo of the drumloop could be controlled by adjusting the playback rate of `PlayBuf`.

```
SynthDef(\drumloopOnsets,{
|rfreq = 1000,modfreq=5,carfreq=100,rate = 1|
var sig, chain, onsets, pips, synthsig,env,tactout,audout;
sig = PlayBuf.ar(1,
       ~drumBuffer, BufRateScale.kr(
              ~drumBuffer),
       loop: 1,rate:rate);
audout = RLPF.ar(sig,rfreq);
chain = FFT(~onsetBuffer, sig);
onsets = Onsets.kr(chain, 0.7, \mkl);
env = EnvGen.kr(Env.perc(0.01,0.2),gate: onsets);
synthsig = SinOsc.ar(carfreq) * SinOsc.ar(modfreq,mul:0.5,add:0.5);
synthsig = synthsig*env;
tactout = SOS.ar(
  SOS.ar(synthsig,0.874225,-1.711427,0.838289,1.711427,-0.712514),
    0.980631,-1.922495,0.941894,1.922495,-0.922526);
    Out.ar(0, [Out.ar(0,audout*0.5), Out.ar(1,tactout * ~tactileGain)]);
}).add;
```

Code 3.4 SuperCollider code of the synth that extracts onsets from a drumloop.

3.8.1 Motion→sound

In Figure 3.13, the motion to sound mapping is represented visually. Similar to Prototype 1, sound was triggered by moving the hand into the trigger zone. In this case a drumloop was triggered. The drumloop kept playing as long as the performer kept the hand within the trigger-zone. While moving the hand along the x-axis, the center frequency of the resonant low pass filter could be controlled continuously. The highest frequency of the filter was mapped to the position that was closest on the picture on the x-axis, while the lowest frequency was mapped to the other end of the axis. The marker z-position controlled the playback rate of the drumloop, mapped low–high to -1–2, where 1 is equal to original playback speed. Thus, when holding the hand at the highest, the drumloop was played back twice as fast. While holding the hand low the playback would be of a negative value, meaning that the drumloop was played backwards. Back and forth motion on the z-axis could therefore result in sounds similar to those produced in DJ-scratching.

3.8.2 Motion→vibrotactile feedback

In the case of Prototype 2, the following are focal points for the vibrotactile feedback:

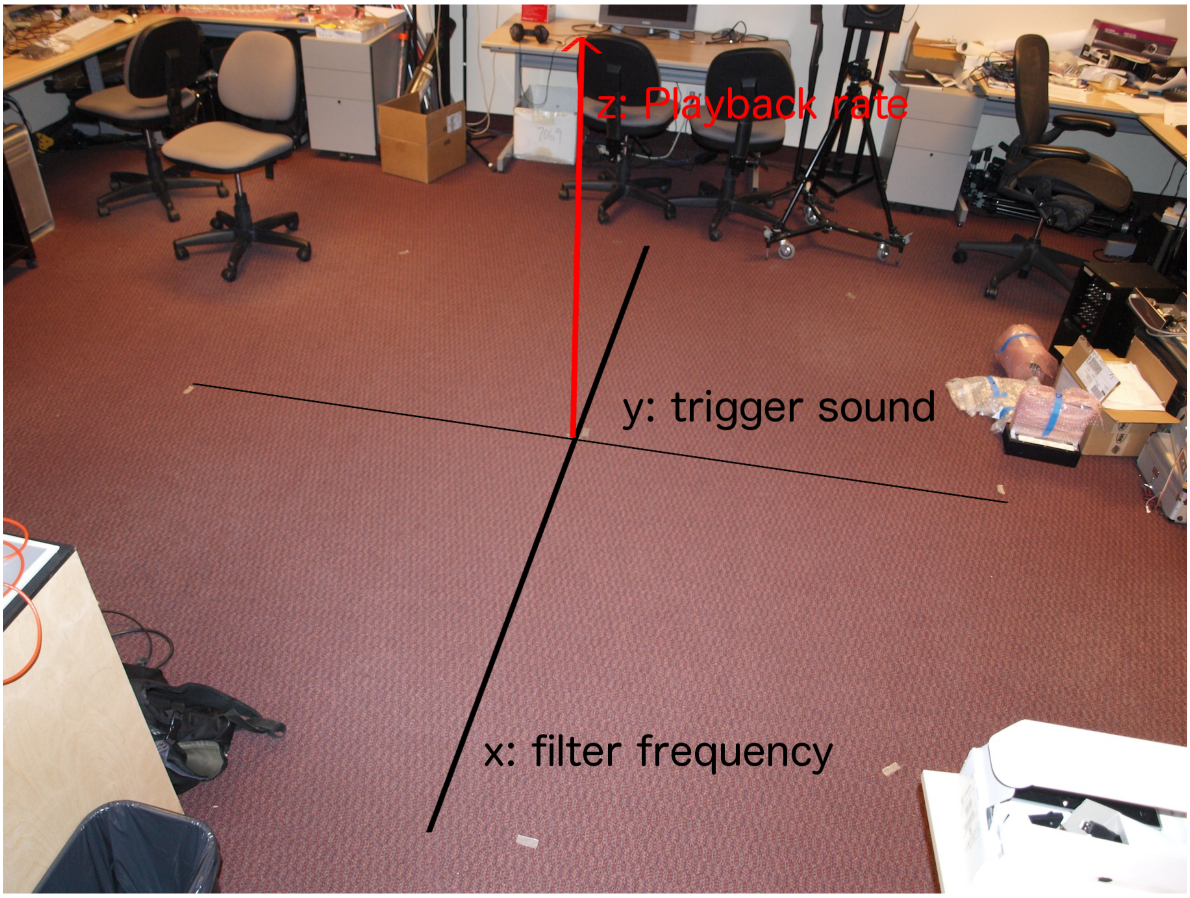


Figure 3.13 The y-axis was used to trigger the drumloop, the x-axis was mapped to the resonant frequency of the filter while the z-axis was mapped to the playback rate of the drumloop.

- Loop triggering: Sound producing action.
- Tempo: Sound modifying action.
- Filter center frequency: Sound modifying action.

Two different feedback strategies where created.

Feedback Strategy 1: Onsets

The first strategy involved extracting onsets of the drumloop that was being played back in the SuperCollider synth (Code 3.4). Onset detection is "the process of detecting the beginning of 'events' such as musical notes in an audio stream" (Stowell and Plumbley, 2007, p. 1). The detected onsets were used to trigger exponentially decaying amplitude envelopes for an amplitude modulated signal.

Depending on the playback rate, the onsets would occur faster or slower. Therefore the iterations of the vibrotactile signal provided rhythmic cues, as well as feedback on the selected playback rate. The z-axis controlled the modulation frequency of the amplitude modulation from low–high to slow–fast, much similar to strategy 3 in prototype 1. This means that the iterated amplitude modulated signal was felt as pulsating with a low playback rate, while it was felt as having a high degree of roughness at a higher playback rate. The carrier frequency of the amplitude modulated signal was controlled by the x-position, meaning that is varied in accordance with the selected frequency of the resonant filter.

Feedback Strategy 2: Filtered sound

This feedback strategy was similar to feedback strategy 4 found in Prototype 1, namely that of providing the filtered audio signal to the actuators.

3.8.3 Example of Interaction

Both feedback strategies provided feedback on the parameters of the sound generator. Similar to Figure 3.11, Figure 3.14 shows en example of interaction with Prototype 2 and feedback strategy 1 (Knutzen et al., 2013). The upper spectrogram in shows the signal created with the feedback strategy.

The spectrogram in the middle shows the audio signal of the drumloop. From time 6–10 seconds one can see how the carrier frequency of the amplitude modulated vibrotactile signal drops along with the the marker x-position. This means that the

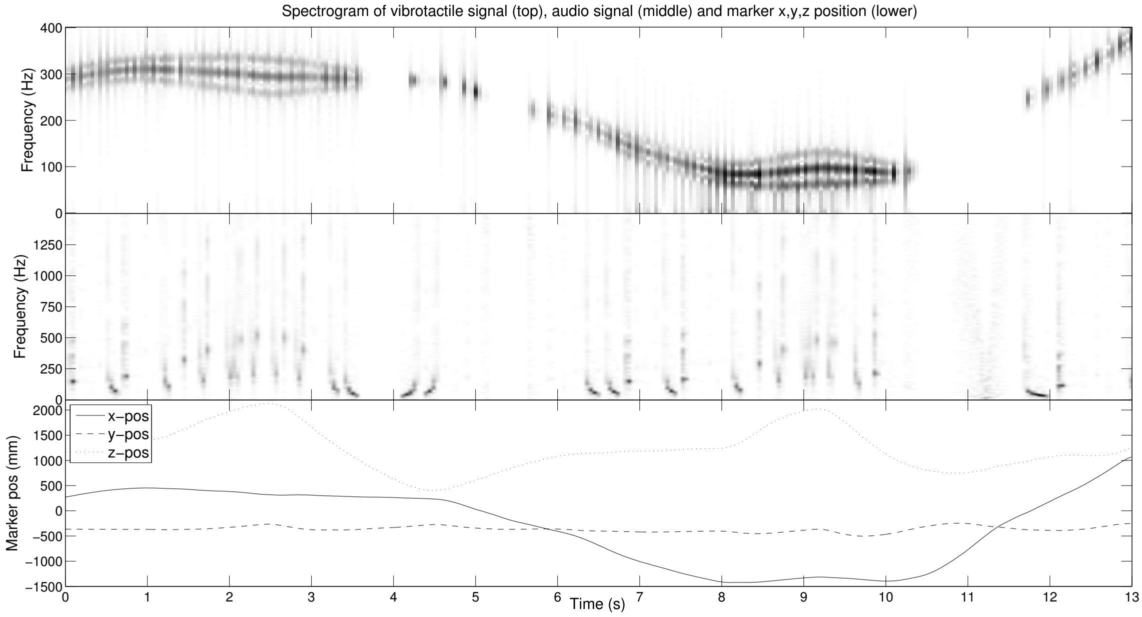


Figure 3.14 This plot shows vibrotactile strategy 1 of prototype 2. The upper plot shows the spectrogram of the amplitude modulated signal, while the middle plot shows the spectrogram of the audio signal. In the bottom plot the marker position as plotted over time.

vibrotactile feedback is varying in accordance with the center frequency of the resonant filter. From time 8–10 seconds one can see a bump in the z-position. During this time the playback rate is at the maximum, i.e. a playback speed of twice the original. The resulting feedback signal is shown as multiple iterations of the vibrotactile signal. The spread in the spectrum also shows that the signal has a high modulation frequency, meaning that the perceived outcome of the signal is a high degree of roughness.

3.9 Summary

With basis in the theory presented in Chapter 2 and the assessment of the hardware and software, two DMI prototypes were constructed. To illustrate how vibrotactile feedback relates to the diverse ways in which on may control sound in DMIs, the two DMI prototypes were constructed to emphasize different approaches to musical control of sound (skill-based/model-based).

Different technology was considered for sensing motion of the performer. The Vicon V460 MoCap system was selected since it can capture motion of the performer's hand in a large space with precision and accuracy. Here the marker position data of the middle finger was used to control sound. Two different vibrotactile actuators were considered and experimented with, namely voice coil actuators and vibration motors. The voice coil actuators were finally chosen since they offer a larger bandwidth with respect to synthesized stimuli signals in the temporal domain. Low latency of the vibrotactile stimuli was considered as more crucial than a wireless implementation. The frequency response of the Sparkfun amplifier was obtained using exponentially swept sinusoidal signals. Here it was revealed that no filtering was needed to compensate, since the frequency response showed a satisfying result.

The musical programming environments Pd, Max MSP and SuperCollider were used to explore control and synthesis of vibrotactile signals, but also sound synthesis. By elaborating on the mapping strategies, the action-sound relationships of the prototypes were explained. Also, the different action-tactile relationships, that is, the vibrotactile feedback strategies, were explained with basis in the theory presented in Chapter 2. To present different approaches, some of the strategies were directly related to the audio signal, while other strategies were related indirectly to the parameters of the sound synthesis.

Chapter 4

Evaluation

I chose to conduct an informal evaluation of the two DMI prototypes developed in Chapter 3 to gain insight in the perceived outcome of the vibrotactile feedback. Since there are several disciplines involved in the field of DMIs, there also exist different approaches to evaluation methods. The different methods assess various aspects related to the DMI design, as well as the experience of playing with the DMI. I will in this chapter first provide an overview of some evaluation methods that are related to evaluation and testing of DMIs. Second, I will present the informal evaluation of the vibrotactile feedback strategies defined in Chapter 3 in light of the presented theory.

4.1 Evaluation Methodology

By applying methods for standardizing the performance of input devices such as *Fitt's Law*, one may compare their performance. This is a well known approach in the field of HCI (Card et al., 1991). Fitt's Law is one method for quantifying the performance of a device related e.g. to the time needed to complete a defined task with the given input device. While input devices are general purpose devices that aer often used for office work, Wanderley and Orio (2002) proposed how DMIs can be evaluated by "borrowing tools from HCI". This entails involving a musical task in the evaluation process. They propose to quantify DMIs in terms of *learnability*, *explorability*, *feature controllability*, and *timing controllability*.

A generalized notion of the performance of input devices with respect to defined tasks in DMIs may be beneficial, for example for DMI builders in the process of choosing input devices for a DMI design. However, Stowell et al. (2009) pointed out that HCI evaluation methods alone are not always applicable when wanting to evaluate DMIs.

They approach the task of evaluating a DMI by using both a qualitative method (discourse analysis) and a quantitative method (Turing test).

Kiefer et al. (2008) approached the task of evaluating a DMI with both qualitative and quantitative analysis. This entailed both analysis of the position data obtained from a WiiMote that was used as the DMI controller, as well as analysis of interviews with the performers. Ghamsari et al. (2013) let the participants of their research focus on improvisation with a novel DMI instead of performing simple defined tasks. Afterwards, through the analysis of in-depth interviews with the participants, their opinion on the mapping strategies of the DMI were revealed.

4.2 Informal Evaluation

With respect to the mentioned methods for evaluation of DMIs, one can see that there are approaches that targets different facets of a DMI. Maintaining a multidisciplinary approach to the thesis problem, HCI approaches for evaluation of the outcome of the vibrotactile feedback strategies in the DMI prototypes were not pursued. Inspired by Ghamsari et al. (2013), I chose an approach where the participants were instructed to play with and explore the DMI prototypes and before answering questions related to perceived feedback. This approach can be seen as an initial step towards revealing tendencies related to the perceived outcome of the vibrotactile feedback strategies. With respect to the thesis problem, the following questions were formulated:

- Was the note selection feedback considered useful?
- Could the participants feel variance in the different vibrotactile feedback signals when varying the control parameters of sound synthesis?
- Did the participants prefer vibrotactile feedback compared to having no feedback at all?
- Did glove design feel obtrusive or limiting when playing the DMI?

4.2.1 Procedure

The evaluation was conducted in the IDMIL at McGill University. Five graduate students from the lab participated without getting paid. All the students are familiar with new DMIs and musical practice. With Prototype 1, the participants were asked to explore the DMI with the instructions to select and trigger different notes. They were also instructed to vary the amplitude and timbre. Similarly, the participants were instructed to explore Prototype 2 and vary the parameters of the sound, i.e. the playback rate and the filter center frequency. For both DMI prototypes, the action-sound relationships were explained before they would start playing. The vibrotactile feedback strategies were not explained to the participants.

The participants first tried Prototype 1. The feedback strategies were presented separately in different orders for each participant. The note grid spread of Prototype 1 was adjusted when switching between vibrotactile strategies. The participants were not told about the change in the grid spread. The rationale behind this decision was to make the role of the note selection feedback more pronounced.

With Prototype 1, the participants were asked to explore the DMI in five separate turns, four turns with vibrotactile feedback and one without any feedback at all. Between each trial the participants were asked whether they could perceive variance in the vibrotactile feedback with respect to their manipulation of the amplitude and timbre of the sound synthesis. Each trial lasted no longer than three minutes. After completing the trials for each prototype, the participants were asked if they preferred having feedback or not, as well as whether they felt that the glove design was obtrusive.

After trying Prototype 1, the participants tried Prototype 2. Here they were also asked whether they could feel that the vibrotactile feedback varied in accordance with their manipulation of the control parameters, that is, triggering the drumloop, controlling the filter frequency of the resonant low pass filter, and the playback rate of the drumloop. Here too, the same questions on preferred feedback and whether or not the glove design felt obtrusive were asked prior to testing both strategies.

4.2.2 Results

Table 4.1 Results of the evaluation of Prototype 1.

Strategy	Amplitude	Timbre
1. Sine	3/5	1/5
2. Burst	2/5	1/5
3. Amplitude modulation	5/5	4/5
4. Filtered audio	4/5	4/5
Question	**Yes**	**No**
Note selection feedback useful?	4	1
Vibrotactile feedback preferred?	5	0
Glove design obtrusive?	1	4

The answers to the questions of the evaluation of Prototype 1 are shown in Table 4.1. An obvious regard here is the low number of participants. Nevertheless, some tendencies related to the of the outcome of the vibrotactile feedback was pointed out. The goal was not to evaluate which of the feedback strategies that is the most successful.

Feedback strategy 1 in Prototype 1 which involved merely a sine wave was perceived by one participant to be varying with the timbre. I find this to be an interesting observation, since there was nothing in the feedback signal that actually varied with the timbre of the sound. Strategy 3 was, judging from the results in Table 4.1, hardest to couple with amplitude and timbral variance. This is perhaps not a surprise, since the vibrotactile signal is very different from the audio signal itself. The relationship between the signal of vibrotactile strategy 3 and the audio signal does not share similarities with respect to e.g. the temporal envelope.

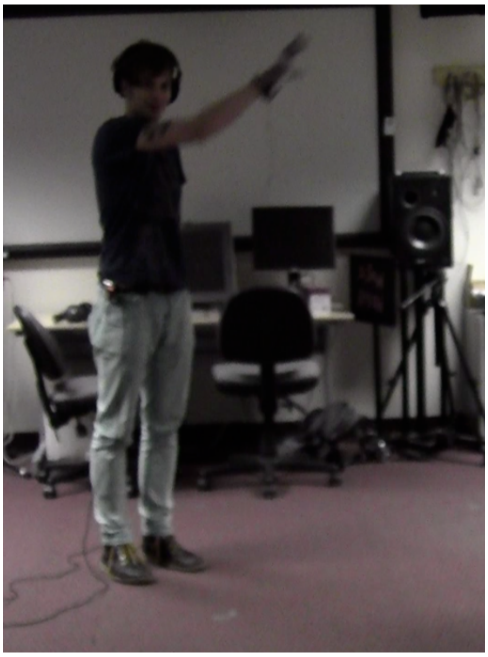


Figure 4.1 One of the participants playing with the DMI prototypes.

The two strategies where most of the participants could feel variance with respect to the varied parameters of the audio synthesis were vibrotactile strategy 3 and 4. Strategy 3 is, as mentioned, based on an amplitude modulated signal where the carrier frequency is $f_0/2$ of the audio signal and the modulation frequency is related low–high to "dull–bright". Strategy 4 is directly related to the sound synthesis since the vibrotactile signal is, in fact, the filtered audio signal.

The response to the questions related to the evaluation of Prototype 2 are shown in

Table 4.2 Results of the evaluation of Prototype 2.

Strategy	Filter frequency	Playback rate
1. Onsets triggered Amplitude modulation	4/5	5/5
2. Filtered audio	1/5	4/5
Question	Yes	No
Vibrotactile feedback preferred?	5	0
Glove design obtrusive?	1	4

Table 4.2. The answers suggest that perceived variance in the feedback with respect to the center frequency of the resonant filter was more prominent with feedback strategy 1.

4.3 Discussion

Rovan and Hayward (2000) and Askenfelt and Jansson (1992) pointed to the importance of vibrotactile feedback for expert performers. In this respect an important consideration is that the participants in the evaluation are all novice performers. The role of vibrotactile feedback may found to have a different function for expert performers.

The motivation for the informal evaluation was to address the research questions further through questioning of the participants' perceived outcome of the vibrotactile feedback for Prototype 1 and 2. These two prototypes are representatives of both skill-based and rule-based DMIs. Thus, the DMI prototypes represent some of the possible ways one may control musical sound when playing DMIs.

While all the feedback strategies of prototype 1 do contain information on the fundamental of the audio signal, the participants were not asked if they could feel a harmonic relationship between the vibrotactile feedback and the audio. This is because absolute frequency discrimination in tactile sensing is poor. However, with Prototype 2, the participants were asked if they could feel variance in the vibrotactile feedback relative to the filter frequency. The fundamental of the vibrotactile signal was varied with the filter frequency. Judging from the answers here it seemed like the synthesized signal was more understandable for conveying information on the filter frequency.

With both Prototypes, all the participants preferred having vibrotactile feedback. There may be different reasons for preference of vibrotactile feedback. One reason is the feeling of the DMI. One of the participants stated that vibrotactile feedback increased the immersive experience of playing with the DMIs. Another participant stated that feedback on the amplitude of the note in Prototype 1 was useful for understanding the velocity of the hand. Conversely, one of the participants claimed to rely more on proprioception than on vibrotactile feedback.

According to the replies of the participants, one may convey information on amplitude, timbre, note triggering, note selection, control of the filter frequency, and playback rate using different vibrotactile strategies. As explained in Chapter 3, the vibrotactile signals may be linked directly to the sound signal (feedback strategy 4 of prototype 1, and feedback strategy 2 of prototype 2), or indirectly (e.g. feedback strategy 3 of prototype 1).

Chapter 5

Conclusions and Discussion

In the introduction of the thesis, the role of haptic feedback in acoustic musical instruments was explained. Here, it was established that, in some cases, the performer relies on haptic feedback (i.e. both kinesthetic and tactile feedback) from the musical instrument. The outcome of such feedback may be crucial for accurate and precise control of the musical instrument. In addition, haptic feedback may also affect the "feel" of the musical instrument.

Since DMIs do not provide haptic feedback inherently, they are lacking many of these qualities that traditional musical instruments provide. DMIs can be controlled using various controllers. One such special case is through the use of open air controllers, which utilize motion in open air to control sound. With open air controllers, the performer may not be touching a physical surface, making such controllers more prone to the issues related to lack of haptic feedback. The particular issue of vibrotactile augmentation of such DMIs was stressed by Rovan and Hayward (2000). In the thesis, this problem was pursued further.

I chose to address the thesis problem with a theoretical and practical approach. First, a theoretical assessment of the thesis problem was provided. Second, a practical implementation with respect to the presented theory was pursued and an informal evaluation was conducted to further investigate the thesis problem. I will in this chapter provide a summary of the thesis content as well a discussion and conclusions.

5.1 Theory and Constraints

Several constraints with respect to theory and research design were employed to keep the content within the scope of a master thesis. At the same time the thesis scope is multidisciplinary with emphasis on the musical aspect of the research questions.

One constraint was the choice to position the thesis research with respect to embodied music cognition. Embodied music cognition bridges the traditional divide between the human corpus and the mind in the investigation of questions related to music. Accordingly, central terms within embodied cognition were presented, namely motion, action, and gesture. These terms are important in embodied music cognition because they stress the embodied participation in music listening and practice. Thus, the thesis topic was presented with regards to a specific direction within musicology.

The distinction *vibro*tactile denotes a subgroup of the umbrella term haptic, i.e. perception of vibration sensation mainly through the mechanoreceptors in the skin. The focus on vibrotactile feedback is therefore also a constraint in that other aspects related to haptic sensing, such as kinesthetic feedback (force feedback), were not pursued in the thesis research. Multimodal perception and tactile perception with respect to physiology was presented with respect to the hands. That is, the emphasis on the hands served as another constraint. Ways of providing information on musical parameters through tactile sensing was explained. This involved a comprehensive explanation of vibrotactile stimuli in both the temporal and the spatial domain.

Since the constituent parts of a DMI are computers and sensors, the involved technical disciplines are for instance HCI. These areas of research need to be taken into account when dealing with DMIs.However, Cook (2001) pointed out that: "Musical interface construction proceeds as more art than science, and possibly this is the only way that it can be done" (p. 4). With this in mind I have tried to balance the involved disciplines with emphasis on the musical aspects.

I provided an explanation of the general architecture of DMIs. In this respect, it was important to emphasize the difference between acoustic musical instruments and DMIs. This is because it is the very source to the problem of the thesis, namely the inherent absence of haptic feedback in DMIs. Along with the explanation, methods for augmenting DMIs with vibrotactile feedback was presented with emphasis on integration of two different kinds of actuators. The problems related to mapping was explained.

The approach of embodied music cognition was also stressed in the explanation of DMIs. In this respect the difference between action-sound relationships and action-sound couplings was pointed out. Since DMIs do not produce sound as a result of the coupling of mechanical objects, they can only incorporate action-sound relationships. Similarly, in DMIs action-tactile relationships do not share the same robustness as in interaction with mechanical objects. The relationships between motion, action and gesture, and sound and vibrotactile stimuli are therefore determined by the mapping strategies. Thus, vibrotactile stimuli can be considered feedback as a result of the mapping strategy.

5.2 Addressing the Research Questions

The research questions of this thesis were the following:

- How can vibrotactile augmentation be implemented in a DMI design with an open air music controller?

    1. What musical information can be conveyed with vibrotactile feedback?
    2. Can vibrotactile augmentation be useful in the context of playing the given DMI?

Given the constraints I elaborated on above, I did not try to approach the implementation in Chapter 3 from a general point of view. With the exception of the Theremin, there are very few established conventions for musical instruments, let alone DMIs, that are based on open air controllers. To exemplify how musical sound can be controlled with such DMIs, two prototypes were constructed. Here sound was controlled using optical infrared marker based MoCap. More specifically, marker position data from one marker on the middle finger was used to obtain motion of the right hand. This motion mapped to control of sound in two different DMI approaches, namely skill-based and model-based approaches.

An overview of previous attempts of haptic augmentation of DMIs was presented in Chapter 2. Here different approaches and technologies were presented as well. With basis in previous research, two low cost actuators that require little complicated equipment were chosen. An assessment of the actuators with respect to vibrotactile signal production and integration in a controller was provided. Focus was then directed on voice coil actuators similar to the ones Egloff (2011) used. The DMI prototypes were augmented with vibrotactile feedback after fitting the voice coil actuators on the inside of the fingertips of the glove controller. This was, in other words, the specific answer to the main research question.

The first subquestion was partly addressed in Chapter 2. Here, I presented an overview of musical parameters that can be conveyed through vibrotactile stimuli. Two main domains were presented, namely the temporal domain and the spatial domain. Belonging to the temporal domain are pitch, amplitude, rhythm, timbre, and roughness.

These were elaborated on with respect to how they occur in musical contexts and how they can be implemented in vibrotactile designs using actuators. In Chapter 3, a brief assessment of vibrotactile feedback in the spatial domain was presented before focusing on feedback in the temporal domain. For Prototype 1 and 2, different strategies were then developed to provide feedback on sound modifying and sound producing actions.

To address the last subquestions further, an informal questioning of the participants in an evaluation was conducted. Here the participants were asked if they could perceive a relationship between the vibrotactile feedback and the musical parameters they were varying, as well as whether they found the feedback to be useful. With respect to subquestion 1, the informal questioning indicates that the participants could feel a relationship between the vibrotactile feedback and the note selection, note triggering, amplitude and timbre of the sound. The feedback strategies do provide feedback on the pitch of the sound. I did however not pursue this matter in the evaluation since frequency discrimination in tactile sensing is poor.

With respect to subquestion 2, one may argue from my approach to the thesis problem, that vibrotactile feedback is useful when playing the DMI prototypes. The informal questioning of the participants suggested that the participants preferred having vibrotactile feedback over no feedback. This may be related to the aspect of feeling in musical instruments. I would also suggest that vibrotactile feedback can be useful since the informal questioning revealed perceived relationships between the vibrotactile stimuli and the manipulation of musical parameters.

5.3 Discussion and Future Work

The informal evaluation in Chapter 4 provided some preliminary results that may inspire future studies. Given the brief overview of evaluation methodology, the vibrotactile feedback strategies may be evaluated with respect to the methodology presented in Chapter 4 (e.g. HCI inspired methodology). While only the marker on the middle finger was taken into account when controlling sound in the DMI prototypes, there exist more advanced ways of to analyze and categorize the motion of the performer using MoCap (Müller, 2007; Mamedes et al., 2013). For future work, such methods may be implemented. This allows the performers to express themselves more extensively.

The influence of tactile sensing in expert performance was pointed out in the thesis. The participants in Chapter 4 were novice performers. In this respect, I will point to areas where tactile sensing may have an effect, namely pedagogy and longevity. For instance, one may study the influence of vibrotactile feedback over a longer time span such that the performers can develop tactile expertise (Harris et al., 2001). With respect to music pedagogy, Giordano and Wanderley (2011) presented how vibrotactile feedback can be used in a learning interface.

Like Brown et al. (2005) pointed out, amplitude modulation could be used to convey information through degree of roughness. Similarly, roughness created by amplitude modulation was used to convey information on musical parameters in both Prototype 1 and 2. While roughness seemed to work adequately for conveying musical information, other some feedback strategies were harder for the participants to couple with parameters of the sound. Thus, for future research it may be interesting to investigate approaches to vibrotactile signal synthesis that may create more understandable and intuitive vibrotactile feedback strategies. From an embodied perspective one can further investigate the influence of action-tactile relationships with respect to action-sound relationships by e.g. developing new approaches to vibrotactile signal synthesis and mapping strategies.

Other areas of research related to vibrotactile feedback is sensory substitution (Egloff, 2011). This means that vibrotactile stimuli can be used to replace a sensing modality that has been disabled. This may be an area of interest for music therapists. Magee and Burland (2008) addressed the issue of lack of haptic feedback in electronic musical instruments that are used in music therapy. A DMI where open air motion is used to control sound is the SoundBeam. This DMI is used frequently in music therapy. Although Prototype 1 and 2 are implemented in a specific environment, with a specific controller, the approach can be ported to other environments and controllers such as the Soundbeam as well. That is, the approach in this thesis may be an area of interest for performers using new open air based DMIs.

The issues of a wireless system was pointed out in the thesis. This problem was not pursued, because of time constraints, but also because it offers a different set of problems than the thesis questions. That being said, for future work this is an area for improvement. This would make the augmentation less obtrusive. Less obtrusiveness would allow more freedom with respect to musical motion, action, and gestures. Thus, the system would become more attractive for the performers. Future work may also entail using different actuators. While the frequency response of the Sparkfun amplifier was measured, future work may involve obtaining the frequency response of the actuators to investigate if inverse filtering is needed.

Bibliography

Askenfelt, A. and Jansson, E. V. (1992). On vibration sensation and finger touch in stringed instrument playing. *Music Perception: An Interdisciplinary Journal*, 9(3):311–349.

Bekkedal, E. (2012). Music kinection : Musical sound and motion in interactive systems. Master's thesis, University of Oslo, Norway.

Berdahl, E., Steiner, H.-C., and Oldham, C. (2008). Practical hardware and algorithms for creating haptic musical instruments. In *Proceedings of the 2008 International Conference on New Interfaces for Musical Expression*, page 61–66, Genova, Italy.

Bevilacqua, F., Müller, R., and Schnell, N. (2005). MnM: a Max/MSP mapping toolbox. In *Proceedings of the 2005 conference on New interfaces for musical expression*, page 85–88, Vancouver, BC.

Bevilacqua, F., Ridenour, J., and Cuccia, D. J. (2002). 3D motion capture data: motion analysis and mapping to music. In *Proceedings of the workshop/symposium on sensing and input for media-centric systems*, Santa Barbara, CA.

Birnbaum, D., Fiebrink, R., Malloch, J., and Wanderley, M. M. (2005). Towards a dimension space for musical devices. In *Proceedings of the 2005 Conference on New Interfaces for Musical Expression*, page 192–195, Vancouver, BC.

Birnbaum, D. M. (2007). Musical vibrotactile feedback. Master's thesis, McGill University, Montréal, QC.

Brinkmann, P. (2012). *Making Musical Apps: Real-time audio synthesis on Android and iOS*. O'Reilly Media, Inc., Sebastopol, CA.

Brinkmann, P., Kirn, P., Lawler, R., McCormick, C., Roth, M., and Steiner, H.-C. (2011). Embedding pure data with libpd. In *Proceedings of the Pure Data Convention*, Weimar, Germany.

Brown, L., Brewster, S., and Purchase, H. (2005). A first investigation into the effectiveness of tactons. In *First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 167–176, Pisa, Italy.

Bryan, N. J., Herrera, J., Oh, J., and Wang, G. (2010). Momu: A mobile music toolkit. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, Sydney, Australia.

Cadoz, C., Lisowski, L., and Florens, J.-L. (1990). A modular feedback keyboard design. *Computer music journal*, 14(2):47–51.

Cadoz, C., Luciani, A., Florens, J.-L., and Castagné, N. (2003). ACROE - ICA: artistic creation and computer interactive multisensory simulation force feedback gesture transducers. In *Proceedings of the 2003 Conference on New Interfaces for Musical Expression*, page 235–246, Montréal, QC.

Card, S. K., Mackinlay, J. D., and Robertson, G. G. (1991). A morphological analysis of the design space of input devices. *ACM Trans. Inf. Syst.*, 9(2):99–122.

Chafe, C. (1993). Tactile audio feedback. In *Proceedings of the International Computer Music Conference*, pages 76–79, Japan.

Chafe, C. and O'Modhrain, S. (1996). Musical muscle memory and the haptic display of performance nuance. In *Proceedings of the International Computer Music Conference*, Hong Kong, China.

Choi, S. and Kuchenbecker, K. J. (2012). Vibrotactile display: Perception, technology, and applications. *Proceedings of the IEEE*, Early Access Online:1–12.

Collins, N. (2010). *Introduction to Computer Music*. John Wiley & Sons, Ltd, Chichester, UK.

Cook, P. (2001). Principles for designing computer music controllers. In *Proceedings of the 2001 conference on New interfaces for musical expression*, page 1–4, Seattle, WA.

Cottle, D. M. (2011). Beginners tutorial. In Wilson, S., Cottle, D. M., and Collins, N., editors, *The SuperCollider Book*, pages 3–54. MIT Press, London, UK.

de Quay, Y., Skogstad, S., and Jensenius, A. (2011). Dance jockey: Performing electronic music by dancing. *Leonardo Music Journal*, (21):11–12.

Dobrian, C. and Bevilacqua, F. (2003). Gestural control of music: using the vicon 8 motion capture system. In *Proceedings of the 2003 Conference on New Interfaces for Musical Expression*, page 161–163, Montréal, QC.

Dourish, P. (2001). *Where the Action is. The Foundations of Embodied Interaction*. MIT Press, Cambridge, MA.

Egloff, D. C. (2011). A vibrotactile music system based on sensory substitution. Master's thesis, Rensselaer Polytechnic Institute, Troy, NY.

Farina, A. (2000). Simultaneous measurement of impulse response and distortion with a swept-sine technique. In *Audio Engineering Society Convention 108*. Audio Engineering Society.

Geldard, F. A. and Sherrick, C. E. (1972). The cutaneous "Rabbit": a perceptual illusion. *Science*, 178(4057):178–179.

Ghamsari, M., Pras, A., and Wanderley, M. M. (2013). Combining musical tasks and improvisation in evaluating novel digital musical instruments. In *Proceedings of the 10th International Symposium on Computer Music Multidisciplinary Research*, pages 506–515, Marseille, France.

Gill, J. (2004). Guidelines for the design of accessible information and communication technology systems. *Royal Institute of the Blind*.

Giordano, M. and Wanderley, M. M. (2011). A learning interface for novice guitar players using vibrotactile stimulation. In *Proceedings of the Sound and Music Computing Conference*, pages 368–373, Padova, Italy.

Giordano, M. and Wanderley, M. M. (2013). Perceptual and technological issues in the design of vibrotactile-augmented interfaces for music technology and media. In Oakley, I. and Brewster, S., editors, *Haptic and Audio Interaction Design*, number 7989 in Lecture Notes in Computer Science, pages 89–98. Springer Berlin Heidelberg.

Godøy, R. I., Haga, E., and Jensenius, A. R. (2006). Exploring music-related gestures by sound-tracing: A preliminary study. *University of Oslo*.

Godøy, R. I. and Leman, M., editors (2010). *Musical Gestures. Sound, Movement, and Meaning*. Routledge, New York, NY.

Goff, G. D. (1967). Differential discrimination of frequency of cutaneous mechanical vibration. *Journal of Experimental Psychology*, 74(2p1):294—294.

Goodwin, A. W. and Wheat, H. E. (2008). Physiological mechanisms of the receptor system. In Grunwald, M., editor, *Human Haptic Perception: Basics and Applications*, pages 93–102. Birkhäuser Basel.

Grunwald, M. (2008). Haptic perception in anorexia nervosa. In Grunwald, M., editor, *Human Haptic Perception: Basics and Applications*, pages 335–351. Birkhäuser Basel.

Gunther, E. and O'Modhrain, S. (2003). Cutaneous grooves: composing for the sense of touch. *Journal of New Music Research*, 32(4):369–381.

Halata, Z. and Baumann, K. I. (2008). Anatomy of receptors. In Grunwald, M., editor, *Human Haptic Perception: Basics and Applications*, pages 85–92. Birkhäuser Basel.

Halmrast, T., Guettler, K., Godøy, R. I., and Bader, R. (2010). Gesture and timbre. In Godøy, R. I. and Leman, M., editors, *Musical Gestures. Sound, Movement, and Meaning*, pages 183—211. Routledge, New York, NY.

Harris, J. A., Harris, I. M., and Diamond, M. E. (2001). The topography of tactile learning in humans. *The Journal of Neuroscience*, 21(3):1056–1061. PMID: 11157091.

Hayward, V. (2008). A brief taxonomy of tactile illusions and demonstrations that can be done in a hardware store. *Brain Research Bulletin*, 75(6):742–752.

Helbig, H. B. and Ernst, M. O. (2008). Haptic perception in interaction with other senses. In Grunwald, M., editor, *Human Haptic Perception: Basics and Applications*, pages 235–249. Birkhäuser Basel.

Hodson, H. (2013). Leap motion hacks show potential of new gesture tech. *New Scientist*, 218(2911):21.

Hunt, A. and Kirk, R. (2000). Trends in gestural control of music. chapter Mapping Strategies for Musical Performance (reprint), pages 231–258. IRCAM, Centre Pompidou, Paris, France.

Hunt, A. and Wanderley, M. M. (2002). Mapping performer parameters to synthesis engines. *Organised Sound*, 7(02):97–108.

Hunt, A., Wanderley, M. M., and Paradis, M. (2003). The importance of parameter mapping in electronic instrument design. *Journal of New Music Research*, 32:429–440.

Jensenius, A. R. (2013). An Action–Sound approach to teaching interactive music. *Organised Sound*, 18(Special Issue 02):178–189.

Jensenius, A. R., Wanderley, M. M., Godøy, R. I., and Leman, M. (2010). Musical gestures. concepts and methods in research. In Godøy, R. I. and Leman, M., editors, *Musical Gestures. Sound, Movement, and Meaning*, pages 12—35. Routledge, New York, NY.

Jordà, S. (2004). Instruments and players: Some thoughts on digital lutherie. *Journal of New Music Research*, 3:321–341.

Kiefer, C., Collins, N., and Fitzpatrick, G. (2008). HCI methodology for evaluating musical controllers: A case study. In *Proceedings of the 2008 Conference on New*

Interfaces for Musical Expression, Genova, Italy.

Kim, Y., Cha, J., Ryu, J., and Oakley, I. (2010). A tactile glove design and authoring system for immersive multimedia. *IEEE MultiMedia*, 17(3):34–45.

Knutzen, H., Kvifte, T., and Wanderley, M. M. (2013). Vibrotactile augmentation of an open-air music controller. In *Proceedings of the 10th International Symposium on Computer Music Multidisciplinary Research*, pages 54–63, Marseille, France.

Kruger, L. (1996). *Pain and Touch.* Academic Press.

Kvifte, T. (2007). *Instruments and the Electronic Age.* Taragot Sounds, Oslo, Norway.

Kvifte, T. (2008). On the description of mapping structures. *Journal of New Music Research*, 37(4):353—362.

Lartillot, O., Toiviainen, P., and Eerola, T. (2008). A matlab toolbox for music information retrieval. In Preisach, C., Burkhardt, P. D. H., Schmidt-Thieme, P. D. L., and Decker, P. D. R., editors, *Data Analysis, Machine Learning and Applications*, Studies in Classification, Data Analysis, and Knowledge Organization, pages 261–268. Springer Berlin Heidelberg.

Leman, M. (2008). *Embodied Music Cognition and Mediation Technology.* MIT Press, Cambridge, MA.

Lim, S.-C., Kwon, D.-S., and Park, J. (2012). Tactile apparent motion between both hands based on frequency modulation. In Isokoski, P. and Springare, J., editors, *Haptics: Perception, Devices, Mobility, and Communication*, number 7282 in Lecture Notes in Computer Science, pages 293–300. Springer Berlin Heidelberg.

Loy, G. (2006). *Musimathics. The Mathematical Foundations of Music*, volume 1. MIT Press, Cambridge, MA.

Machover, T. and Chung, J. (1989). Hyperinstruments: Musically intelligent and interactive performance and creativity systems. In *Proceedings of the International Computer Music Conference*, pages 186–190, Ohio State University, OH.

Magee, W. L. and Burland, K. (2008). An exploratory study of the use of electronic music technologies in clinical music therapy. *Nordic Journal of Music Therapy*, 17(2):124–141.

Malloch, J., Birnbaum, D., Sinyor, E., and Wanderley, M. M. (2006). Towards a new conceptual framework for digital musical instruments. In *Proceedings of the International Conference on Digital Audio Effects*, pages 49—52, McGill University, Montréal, QC.

Malloch, J., Sinclair, S., Hollinger, A., and Wanderley, M. M. (2011). Input devices and music interaction. In *Musical Robots and Interactive Multimodal Systems*, page 67–83. Springer.

Malloch, J., Sinclair, S., and Wanderley, M. M. (2013). Libmapper: (a library for connecting things). In *CHI '13 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '13, page 3087–3090, New York, NY, USA. ACM.

Mamedes, C. R., Wanderley, M. M., Manzolli, J., and Garcia, D. H. L. (2013). Strategies for mapping control in interactive audiovisual installations. In *Proceedings of the 10th International Symposium on Computer Music Multidisciplinary Research*, pages 766–778, Marseille, France.

Marshall, M. T. and Wanderley, M. M. (2006). Vibrotactile feedback in digital musical instruments. In *Proceedings of the 2006 conference on New interfaces for musical* *expression*, page 226–229, Paris, France.

Martin, A. G. V. (2011). Touchless gestural control of concatenative sound synthesis. Master's thesis, McGill University, Montréal, QC.

Mathews, M. (2001). What is loudness? In Cook, P. R., editor, *Music, Cognition and Computerized Sound*. MIT Press, Cambridge, MA.

Mathews, M. V. (1991). The radio baton and conductor program, or: Pitch, the most important and least expressive part of music. *Computer Music Journal*, 15(4):37–46.

Miranda, E. R. and Wanderley, M. M. (2006). *New Digital Musical Instruments: Control and Interaction Beyond the Keyboard*. A-R Editions, Inc, Middleton, WI.

Miyazaki, M., Hirashima, M., and Nozaki, D. (2010). The "Cutaneous rabbit" hopping out of the body. *The Journal of Neuroscience*, 30(5):1856–1860.

Montague, S. (1991). Rediscovering leon theremin. *Tempo*, (177):18–23.

Moore, F. R. (1990). *Elements of Computer Music*. Prentice Hall, Englewood Cliffs, NJ.

Morris, G. C., Leitman, S., and Kassianidou, M. (2004). SillyTone squish factory. In *Proceedings of the 2004 conference on New interfaces for musical expression*, page 201–202, Hamamatsu, Japan.

Moss, W. and Cunitz, B. (2005). Haptic theremin: Developing a haptic musical controller using the sensable phantom omni. In *Proceedings of the International Computer Music Conference*, pages 275–277, Barcelona, Spain.

Mulder, A. G. E. (2000). Towards a choice of gestural constraints for instrumental performers. In Wanderley, M. M. and Battier, M., editors, *Trends in Gestural Control of Music*, pages 315–335. IRCAM, Centre Pompidou, Paris.

Müller, M. (2007). *Information Retrieval for Music and Motion*. Springer, Berlin Heidelberg.

Nettl, B. (2013). Music. *Grove Music Online*.

Nusseck, M. and Wanderley, M. M. (2009). Music and Motion—How music-related ancillary body movements contribute to the experience of music. *Music Perception: An Interdisciplinary Journal*, 26(4):335–353.

Nymoen, K., Skogstad, S. A. v. D., and Jensenius, A. R. (2011). SoundSaber - a motion capture instrument. In *Proceedings of the 2011 International Conference on New Interfaces for Musical Expression*, pages 312–315, Oslo, Norway.

Oakley, I., McGee, M. R., Brewster, S., and Gray, P. (2000). Putting the feel in "look and feel". In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 415–422, New York, NY. ACM.

Okazaki, R., Hachisu, T., Sato, M., Fukushima, S., Hayward, V., and Kajimoto, H. (2013). Judged consonance of tactile and auditory frequencies. In *Proceedings of the IEEE WHC*, pages 663–666, Daejeon, Korea.

O'Modhrain, M. S. (2001). *Playing by feel: incorporating haptic feedback into computer-based musical instruments*. PhD thesis, Stanford University, Stanford, CA.

O'Sullivan, D. and Igoe, T. (2004). *Physical Computing: Sensing and Controlling the Physical World with Computers*. Cengage Learning, Boston, MA.

Paradiso, J. A. and Gershenfeld, N. (1997). Musical applications of electric field sensing. *Computer Music Journal*, 21(2):69–89.

Park, G. and Choi, S. (2011). Perceptual space of amplitude-modulated vibrotactile stimuli. In *2011 IEEE World Haptics Conference (WHC)*, pages 59–64.

Partan, S. and Marler, P. (1999). Communication goes multimodal. *Science*, 283(5406):1272–1273.

Picinali, L., Feakes, C., Mauro, D. A., and Katz, B. F. G. (2012). Spectral discrimination thresholds comparing audio and haptics for complex stimuli. In Magnusson, C., Szymczak, D., and Brewster, S., editors, *Haptic and Audio Interaction Design*, number 7468 in Lecture Notes in Computer Science, pages 131–140. Springer Berlin Heidelberg.

Pongrac, H. (2008). Vibrotactile perception: examining the coding of vibrations and the just noticeable difference under various conditions. *Multimedia Systems*, 13(4):297–307.

Puckette, M. (2007). *The Theory and Techniques of Electronic Music*. World Scientific, Singapore.

Rizun, P., Gunn, D., Cox, B., and Sutherland, G. (2006). Mechatronic design of haptic forceps for robotic surgery. *The International Journal of Medical Robotics and Computer Assisted Surgery*, 2(4):341–349.

Roads, C. (1996). *The Computer Music Tutorial*. MIT Press, Cambridge, MA.

Rossing, T. D., Moore, F. R., and Wheeler, P. A. (2002). *The Science of Sound*. Addison Wesley, San Francisco, CA, 3rd edition.

Rovan, J. and Hayward, V. (2000). Typology of tactile sounds and their synthesis in gesture-driven computer music performance. In Wanderley, M. M. and Battier, M., editors, *Trends in Gestural Control of Music*, pages 355–368. IRCAM, Centre Pompidou, Paris.

Russo, F. A., Ammirante, P., and Fels, D. I. (2012). Vibrotactile discrimination of musical timbre. *Journal of Experimental Psychology: Human Perception and Performance*, 38(4):822–826.

Sachs, D. M. (2005). A forearm controller and tactile display. Master's thesis, MIT, MA.

Schacher, J. C. (2007). Gesture control of sounds in 3D space. In *Proceedings of the 2007 international conference on New interfaces for musical expression*, pages 358–362, New York, NY.

Schumacher, M., Giordano, M., Wanderley, M. M., and Ferguson, S. (2013). Vibrotactile notification for live electronics performance: A prototype system. In *Proceedings of the 10th International Symposium on Computer Music Multidisciplinary Research*, pages 516–525, Marseille, France.

Sethares, W. A. (2007). *Rhythm and Transforms*. Springer.

Shapiro, L. (2011). *Embodied Cognition*. Routledge, London, UK.

Sinclair, S. (2012). *Velocity-driven Audio-Haptic Interaction With Real-Time Digital Acoustic Models*. PhD, McGill University, Montréal, QC.

Steiner, H.-C. (2006). Towards a catalog and software library of mapping methods. In *Proceedings of the 2006 conference on New interfaces for musical expression*, page 106–109, Paris.

Stowell, D. and Plumbley, M. (2007). Adaptive whitening for improved real-time audio onset detection. In *Proceedings of the International Computer Music Conference*, Copenhagen.

Stowell, D., Robertson, A., Bryan-Kinns, N., and Plumbley, M. D. (2009). Evaluation of live human–computer music-making: Quantitative and qualitative approaches. *International Journal of Human-Computer Studies*, 67(11):960–975.

Sziebig, G., Solvang, B., Kiss, C., and Korondi, P. (2009). Vibro-tactile feedback for VR systems. In *2nd Conference on Human System Interactions, 2009. HSI '09*, pages 406–410.

Thibodeau, J. and Wanderley, M. M. (2013). Trumpet augmentation and technological symbiosis. *Computer Music Journal*, 37(3):12–25.

Vallbo, A. B. and Johansson, R. S. (1984). Properties of cutaneous mechanoreceptors in the human hand related to touch sensation. *Hum Neurobiol*, 3(1):3–14.

Verrillo, R. T. (1992). Vibration sensation in humans. *Music Perception*, page 281–302.

Wanderley, M. and Depalle, P. (2004). Gestural control of sound synthesis. *Proceedings of the IEEE*, 92(4):632–644.

Wanderley, M. M. and Orio, N. (2002). Evaluation of input devices for musical expression: Borrowing tools from HCI. *Computer Music Journal*, 26(3):62–76.

Wanderley, M. M., Vines, B. W., Middleton, N., McKay, C., and Hatch, W. (2005). The musical significance of clarinetists' ancillary gestures: An exploration of the field. *Journal of New Music Research*, 34(1):97–113.

Wessel, D. and Wright, M. (2002). Problems and prospects for intimate musical control of computers. *Computer Music Journal*, 26(3):11–22.

Wilson, S., Cottle, D., and Collins, N., editors (2011). *The SuperCollider Book*. MIT Press, London, UK.

Wright, M. (2005). Open sound control: an enabling technology for musical networking. *Organised Sound*, 10(03):193–200.

Young, G., Murphy, D., and Weeter, J. (2013). Audio-tactile glove. In *Proc. of the 16th Int. Conference on Digital Audio Effects (DAFx-13)*, Maynooth, Ireland.

Zhang, Z. (2012). Microsoft kinect sensor and its effect. *IEEE MultiMedia*, 19(2):4–10.

Appendix A

SuperCollider Code

A.1 Code for Vibration Motor Experimentation

```
// SuperCollider code for creating vibration patterns with circular
// pager motors, libmapper, firmapper and arduino
// licensed under GNU GPL 3

( // instatiate libmapper device and create outputs
a = MapperDevice.new;
b = a.addOutput('/toPwm', 1, $i, 'PWM', 0, 255);
c = a.addOutput('/toPwm2', 1, $i, 'PWM', 0, 255);
d = a.addOutput('/toPwm3', 1, $i, 'PWM', 0, 255);
e = a.addOutput('/toPwm4', 1, $i, 'PWM', 0, 255);
h = a.addOutput('/toPwm5',1,$i,'PWM',0,255);
f = a.addInput('/leftRight', 1, $i,'bool',0,1,{
       arg signame, instanceid, value;
       value.postln; if(value == 0, {~lr.play},{~rl.play});});
g = a.addInput('/tapsIn', 1, $i,'bool',0,1,{
       arg signame, instanceid, value;
       value.postln; if(value == 0, {~bwpc.play},{~pcbw.play});});
~debugIn = a.addInput('/debugIn',1,$i,'PWM',0,255, {
       arg signame, instanceid, value;
       "debug:".post; value.postln;
});
)

( //panic button kill vibration
b.update(0);
c.update(0);
d.update(0);
```

```
e.update(0);
h.update(0);
)
//(
//Env.cutoff(1, 255, \sqr).asSignal(70).round(1).addAll([0,0,0,0,0]).plot('sqrt envelope',discrete: true,minval:0,maxval:255);
//Env.cutoff(1, 255, \sqr).asSignal(70).round(1).reverse.addAll([0,0,0,0,0]).plot('reverse sqrt envelope',discrete: true,minval:0,maxval:255);
//)
(
//~foo = (0..255).add(0); // linear ramp from 0-255
~foo = Env.new([0,255],[1,1],curve:'lin').asSignal(70).round(1).add(0).add(0).add(0);
~percArray = ~foo.reverse;
~globalWait = (1/200); // freq of tactile signal (update rate) should have pwm freq much higher than this
~level = 255; // peaklevel for percussive envelope
//~percArray = Env.perc(level: ~level).asSignal(255).round(1);

~bup = Routine{~foo.do{arg in; b.update(in); ~globalWait.wait}; ~bup.yieldAndReset;};
~cup = Routine{~foo.do{arg in; c.update(in);~globalWait.wait}; ~cup.yieldAndReset;};
~dup = Routine{~foo.do{arg in; d.update(in); ~globalWait.wait}; ~dup.yieldAndReset;};
~eup = Routine{~foo.do{arg in; e.update(in); ~globalWait.wait}; ~eup.yieldAndReset;};
~hup = Routine{~foo.do{arg in; h.update(in); ~globalWait.wait}; ~hup.yieldAndReset;};

~bdown = Routine{~foo.reverse.do{arg in; b.update(in);~globalWait.wait}; ~bdown.yieldAndReset;};
~cdown = Routine{~foo.reverse.do{arg in; c.update(in);~globalWait.wait}; ~cdown.yieldAndReset;};
~ddown = Routine{~foo.reverse.do{arg in; d.update(in); ~globalWait.wait}; ~ddown.yieldAndReset;};
~edown = Routine{~foo.reverse.do{arg in; e.update(in); ~globalWait.wait}; ~edown.yieldAndReset;};
~hdown = Routine{~foo.reverse.do{arg in; h.update(in); ~globalWait.wait}; ~hdown.yieldAndReset;};

~bsdown = Routine{~percArray.do{arg in; b.update(in); ~globalWait.wait}; ~bsdown.yieldAndReset;};
```

```
~csdown = Routine{~percArray.do{arg in; c.update(in); ~globalWait.wait}; ~csdown.yieldAndReset;};
~dsdown = Routine{~percArray.do{arg in; d.update(in); ~globalWait.wait}; ~dsdown.yieldAndReset;};
~esdown = Routine{~percArray.do{arg in; e.update(in); ~globalWait.wait}; ~esdown.yieldAndReset;};
~hsdown = Routine{~percArray.do{arg in; h.update(in); ~globalWait.wait}; ~hsdown.yieldAndReset;};

~bsup = Routine{~percArray.reverse.do{arg in; b.update(in);~globalWait.wait}; ~bsup.yieldAndReset;};
~csup = Routine{~percArray.reverse.do{arg in; c.update(in);~globalWait.wait}; ~csup.yieldAndReset;};
~dsup = Routine{~percArray.reverse.do{arg in; d.update(in); ~globalWait.wait}; ~dsup.yieldAndReset;};
~esup = Routine{~percArray.reverse.do{arg in; e.update(in);~globalWait.wait}; ~esup.yieldAndReset;};
~hsup = Routine{~percArray.reverse.do{arg in; h.update(in); ~globalWait.wait}; ~hsup.yieldAndReset;};

~bwpc = Routine{~bdown.play;~cup.play;~bwpc.yieldAndReset;}; // from bottom wrist to palm center
~pcbw = Routine{~bup.play;~cdown.play;~pcbw.yieldAndReset;}; // from palm center to bottom wrist
~lr = Routine{~edown.play;~dup.play;~lr.yieldAndReset;}; // left to right
~rl = Routine{~eup.play;~ddown.play;~rl.yieldAndReset;}; // right to left
~bwth = Routine{~hup.play;~bdown.play;~bwth.yieldAndReset;}; // bottom wrist to top hand
~thbw = Routine{~hdown.play;~bup.play;~thbw.yieldAndReset;}; // top hand to bottom wrist
~lbwpc = Routine{~bdown.play;~cup.play;~esup;~lbwpc.yieldAndReset;}; // from bottom wrist to palm center + left
~lpcbw = Routine{~esdown;~bup.play;~cdown.play;~lpcbw.yieldAndReset;}; // from palm center + left to bottom wrist
~rbwpc = Routine{~bdown.play;~cup.play;~dsup;~rbwpc.yieldAndReset;}; // from bottom wrist to palm center + right
~rpcbw = Routine{~dsdown;~bup.play;~cdown.play;~rpcbw.yieldAndReset;}; // from palm center + right to bottom wrist

)
// attempt to create apparent motion
//
~bwpc.play; // from bottom wrist to palm center
```

```
~pcbw.play;// from palm center to bottom wrist
~lr.play; // left to right
~rl.play;// right to left
~bwth.play; // bottom wrist to top hand
~thbw.play; // top hand to bottom wrist
~lbwpc.play; // from bottom wrist to palm center + left
~lpcbw.play; // from palm center + left to bottom wrist
~rbwpc.play; // from bottom wrist to palm center + right
~rpcbw.play; // from palm center to bottom wrist
~esdown.play

(
// from palm center to bottom wrist
~bup.play;
~esdown.play;
~cdown.play;
)

(
// from palm center to bottom wrist
~bup.play;
~esdown.play;
~cdown.play;
)

(
// from palm center to bottom wrist
~bup.play;
~esdown.play;
~cdown.play;
)
```

A.2 Prototype 1 and 2

A.2.1 Code for Control

```
// Code licensed under GNU GPL 3
//

// used to change the spread during the test
~boarderDivisor = 300;
(
```

```
// variables used in processing of the input
~boarderFreq = 0;
~croTimb = 0;
~croAmp = 0;
~sawCutoff = 0;
~amMod = 0;
~impFreq = 0;

//libmapper devices
~dev1 = MapperDevice.new(’Vicon’);
~dev2 = MapperDevice.new(’SuperCollider’);

// Vicon outputs
~vic1 = ~dev1.addOutput(’/markerX’, 1, $f, ’pos’, -2000, 2000);
~vic2 = ~dev1.addOutput(’/markerY’, 1, $f, ’pos’, -2000, 2000);
~vic3 = ~dev1.addOutput(’/markerZ’, 1, $f, ’pos’, 300, 1800);
~vic4 = ~dev1.addOutput(’/markerXvelAbs’, 1, $f, ’vel’, 0,10);
~vic5 = ~dev1.addOutput(’/markerYvelAbs’, 1, $f, ’vel’, 0,10);
~vic6 = ~dev1.addOutput(’/markerZvelAbs’, 1, $f, ’vel’, 0,10);

// supercollider libmapper inputs. See json files for mappings
~amModIn = ~dev2.addInput(’/amModFreqCrotale’,1, $f, ’Hz’, 7, 20, {
               |signame, instanceid, value|
               ~amMod = value});
~croTimbIn = ~dev2.addInput(’/crotaleTimbre’,1, $f, ’timb’, 0, 3, {
               |signame, instanceid, value|
               ~croTimb = value;});
~setup4ImpFreq = ~dev2.addInput(’/impBurstFreq’,1, $f, ’Hz’, 3, 15, {
               |signame, instanceid, value|
               ~impFreq = value;});
~croAmpIn = ~dev2.addInput(’/crotaleAmp’,1, $f, ’amp’, 0, 1, {
               |signame, instanceid, value|
               ~croAmp = value;});
~inDebug = ~dev2.addInput(’/inDebug’,1, $f, ’any’,0,1,{
       |signame, instanceid, value|
       postln("debug: " + value);});

~prevDebug = 0;
~croDebug = ~dev2.addInput(’/crotaleDebug’, 1, $f, ’pos’, -2000, 2000,{
       |signame, instanceid, value|
       if(value.isNegative && ~prevDebug.isPositive,
               {postln("croamp: " + ~croAmp + "croTimb: " + ~croTimb + "boarderFreq:
                   " + ~boarderFreq)}
```

```
        ,{});
        ~prevDebug = value;
});

// the function defined in the input triggers the tactile synth whenever a threshold is
// crossed. Also the frequency variable is set so that when the audio synth is
// triggered the frequency of that synth is set equal to ~boarderFreq. Input to this should be
// X-position of marker2.
~prev2 = 0;
~boarderDivisor = 200; // controls the spread of the grid
/*~boarderTactSin = ~dev2.addInput('/boarderSynthTactSin', 1, $f, 'pos', -2000, 2000,{
        |signame, instanceid, value|
        var func,modvalue;
        modvalue = (value/~boarderDivisor).round(1).mod(8);
        if((modvalue - ~prev2) != 0, {case
        {modvalue == 0}{~boarderFreq = 60; Synth(\tactsineSynth,[\freq,60])}
        {modvalue == 1}{~boarderFreq = 63; Synth(\tactsineSynth,[\freq,63])}
        {modvalue == 2}{~boarderFreq = 65; Synth(\tactsineSynth,[\freq,65])}
        {modvalue == 3}{~boarderFreq = 67; Synth(\tactsineSynth,[\freq,67])}
        {modvalue == 4}{~boarderFreq = 70; Synth(\tactsineSynth,[\freq,70])}
        {modvalue == 5}{~boarderFreq = 72; Synth(\tactsineSynth,[\freq,72])}
        {modvalue == 6}{~boarderFreq = 75; Synth(\tactsineSynth,[\freq,75])}
        {modvalue == 7}{~boarderFreq = 77; Synth(\tactsineSynth,[\freq,77])}
        },{});
        ~prev2 = modvalue;
});*/

~boarderTactSin = ~dev2.addInput('/boarderSynthTactImp', 1, $f, 'pos', -2000, 2000,{
        |signame, instanceid, value|
        var func,modvalue;
        modvalue = (value/~boarderDivisor).round(1).mod(8);
        if((modvalue - ~prev2) != 0, {case
        {modvalue == 0}{~boarderFreq = 60; Synth(\tactImpSynthBoarder,[\amp,1]);}
        {modvalue == 1}{~boarderFreq = 63; Synth(\tactImpSynthBoarder,[\amp,1]);}
        {modvalue == 2}{~boarderFreq = 65; Synth(\tactImpSynthBoarder,[\amp,1]);}
        {modvalue == 3}{~boarderFreq = 67; Synth(\tactImpSynthBoarder,[\amp,1]);}
        {modvalue == 4}{~boarderFreq = 70; Synth(\tactImpSynthBoarder,[\amp,1]);}
        {modvalue == 5}{~boarderFreq = 72; Synth(\tactImpSynthBoarder,[\amp,1]);}
        {modvalue == 6}{~boarderFreq = 75; Synth(\tactImpSynthBoarder,[\amp,1]);}
        {modvalue == 7}{~boarderFreq = 77; Synth(\tactImpSynthBoarder,[\amp,1]);}
```

```
	},{});
	~prev2 = modvalue;
});

~prevVal1 = 0;
~setup1 = ~dev2.addInput('/sineOnlyCrotale', 1, $f, 'pos', -2000, 2000,{
	|signame, instanceid, value|
	if(value.isNegative && ~prevVal1.isPositive,
		{Synth(\PMCrotaleMod,[\midi,~boarderFreq,\varpar,~croTimb,\amp,~croAmp,\tactAmp,0]);
		Synth(\tactSin,[\freq,~boarderFreq,\amp,~croAmp]);
	},{});
	~prevVal1 = value;
});

~prevVal2 = 0;
~setup2 = ~dev2.addInput('/sameSoundCrotale', 1, $f, 'pos', -2000, 2000,{
	|signame, instanceid, value|
	if(value.isNegative && ~prevVal2.isPositive,
		{Synth(\PMCrotaleMod,[\midi,~boarderFreq,\varpar,~croTimb,\amp,~croAmp,\tactamp,30]);
	},{});
	~prevVal2 = value;
});

~prevVal3 = 0;
~setup3 = ~dev2.addInput('/amTactCrotale', 1, $f, 'pos', -2000, 2000,{
	|signame, instanceid, value|
	if(value.isNegative && ~prevVal3.isPositive,
		{Synth(\PMCrotaleMod,[\midi,~boarderFreq,\varpar,~croTimb,\amp,~croAmp,\tactAmp,0]);
			Synth(\tactAM,[\carfreq,~boarderFreq,\amp,~croAmp,\modfreq,~amMod]);
	},{});
	~prevVal3 = value;
});

~prevVal4 = 0;
~setup4 = ~dev2.addInput('/impBurstCrotale', 1, $f, 'pos', -2000, 2000,{
	|signame, instanceid, value|
	if(value.isNegative && ~prevVal4.isPositive,
		{Synth(\PMCrotaleMod,[\midi,~boarderFreq,\varpar,~croTimb,\amp,~croAmp,\tactAmp,0]);
```

```
                       Synth(\tactImpSynth,[\imppar,~impFreq,\amp,~croAmp,\freq,~
                           boarderFreq]);
        },{});
        ~prevVal4 = value;
});

~prevVal5 = 0;
~drumbool = false;
~drumloop = ~dev2.addInput('/drumloopPure', 1, $f, 'pos', -2000, 2000,{
        |signame, instanceid, value|
        case
        {value.isNegative && ~prevVal5.isPositive}{a = Synth(\drumloop,[\bufnum,~
            drumBuffer,\loop,1]);~drumbool = true;}
        {value.isPositive && ~prevVal5.isNegative}{a.free; ~drumbool = false;};
        ~prevVal5 = value;
});
~prevVal6 = 0;
~drumbool2 = false;
~drumloop2 = ~dev2.addInput('/drumloopOnsets', 1, $f, 'pos', -2000, 2000,{
        |signame, instanceid, value|
        case
        {value.isNegative && ~prevVal6.isPositive}{a = Synth(\drumloopOnsets,[\loop
            ,1,\bufnum,~drumBuffer,\gain,3]);~drumbool2 = true;}
        {value.isPositive && ~prevVal6.isNegative}{a.free; ~drumbool2 = false;};
        ~prevVal6 = value;
});
~drumResFreq = ~dev2.addInput('/drumResFreq',1,$f,'Hz',50,12000,{
        |signame,instanceid, value|
        if (~drumbool,{a.set(\rfreq,value)},{})
});
~drumResFreq2 = ~dev2.addInput('/drumResFreq2',1,$f,'Hz',50,12000,{
        |signame,instanceid, value|
        if (~drumbool2,{a.set(\rfreq,value)},{})
});
~drumRate = ~dev2.addInput('/drumRate',1,$f,'rate',-1,2,{
        |signame,instanceid, value|
        if (~drumbool,{a.set(\rate,value)},{})
});
~drumRate2 = ~dev2.addInput('/drumRate2',1,$f,'rate',-1,2,{
        |signame,instanceid, value|
        if (~drumbool2,{a.set(\rate,value)},{})
});
```

```
~drumOnsetCarFreq = ~dev2.addInput('/drumOnsetCarFreq',1,$f,'Hz',15,500,{
        |signame,instanceid, value|
        if (~drumbool2,{a.set(\carfreq,value)},{})
});
~drumOnsetModFreq = ~dev2.addInput('/drumOnsetModFreq',1,$f,'rate',1,30,{
        |signame,instanceid, value|
        if (~drumbool2,{a.set(\modfreq,value)},{})
});

~prevVal7 = 0;
~drumbool3 = false;
~drumloop3 = ~dev2.addInput('/drumloopPreview', 1, $f, 'pos', -2000, 2000,{
        |signame, instanceid, value|
        case
        {value.isNegative && ~prevVal6.isPositive}{
                a = Synth(\drumloopPreview,[\loop,1,\bufnum]);
                ~previewSynth.set(\amp,0);
                ~drumbool3 = true;}
        {value.isPositive && ~prevVal6.isNegative}{
                a.free;
                ~previewSynth.set(\amp,1);
                ~drumbool3 = false};
        ~prevVal6 = value;
});

~drumPreviewAm = ~dev2.addInput('/drumPreviewAMCar',1,$f,'Hz',15,400,{
        |signame,instanceid, value|
        if(~drumbool3,~previewSynth.set(\carfreq,value),{});
});

~drumPreviewMod = ~dev2.addInput('/drumPreviewAMMod',1,$f,'Hz',1,17,{
        |signame,instanceid, value|
        if(~drumbool3,~previewSynth.set(\modfreq,value),{});
});

~drumResFreq3 = ~dev2.addInput('/drumResFreq3',1,$f,'Hz',50,12000,{
        |signame,instanceid, value|
        if (~drumbool3,{a.set(\rfreq,value)},{})
});
~drumRate3 = ~dev2.addInput('/drumRate3',1,$f,'rate',-1,2,{
        |signame,instanceid, value|
        if (~drumbool3,{a.set(\rate,value)},{})
});
```

```
)

// trace OSC input
OSCFunc.trace(true);
OSCFunc.trace(false);
// record marker position data
~fileWriteX = File("markerX","w");
~fileWriteY = File("markerY","w");
~fileWriteZ = File("markerZ","w");
~fileWriteYVel = File("markerYVel","w");

(
n = NetAddr.new("192.168.1.101", 1297);
~velPrevY = 0;
// function that responds to Vicon input and sends both the distance value and the
// marker position values to the libmapper device outputs.
OSCdef.new(\test, {|msg, time, addr, recvPort|
        var vel;
        ~vic1.update(msg[1]);~vic2.update(msg[2]);~vic3.update(msg[3]);
        vel = abs(msg[2] - ~velPrevY);
        ~vic5.update(vel);
        if(~writeBool, {
                ~fileWriteX.write(msg[1] + "\n");
                ~fileWriteY.write(msg[2] + "\n");
                ~fileWriteZ.write(msg[3] + "\n");
                ~fileWriteYVel.write(vel + "\n");
        },{});
        ~velPrevY = msg[2];
                                },
        '/cross/Marker2/P',n);
)

// boolean used to trigger recording of mocap data
~writeBool = false;
~writeBool = true;
//closing and freeing files with the mocap data
~fileWriteX.close;
~fileWriteY.close;
~fileWriteZ.close;
~fileWriteYVel.close;
~fileWriteX.free;
~fileWriteY.free;
~fileWriteZ.free;
```

```
~fileWriteYVel.free;

// buffers for recording of audio and tactile signal
~buffer1.free;
~buffer2.free;
~recordLength = 20;
~buffer1 = Buffer.alloc(s, 44100 * ~recordLength,1);
~buffer2 = Buffer.alloc(s, 44100 * ~recordLength,1);

(
~buffer1.write(sampleFormat: 'int24',headerFormat: "aiff");
thisProcess.platform.recordingsDir +/+ "buf1_" ++ Date.localtime.stamp ++ ".aiff";
)

// synths used for recording
SynthDef(\recordBuffersLeft, {
	|buf|
	var in = In.ar(0,1);
	RecordBuf.ar(in,buf,loop:0,doneAction:2);
}).add;
SynthDef(\recordBuffersRight, {
	|buf|
	var in = In.ar(1,1);
	RecordBuf.ar(in,buf,loop:0,doneAction:2);
}).add;

// function to trigger recording of audio, tactile signal and mocap data
fork({
	~writeBool = true;
	Synth(\recordBuffersLeft,[\buf,~buffer1]);
	Synth(\recordBuffersRight,[\buf,~buffer2]);
	20.wait;
	~writeBool = false;
})
```

A.2.2 Vibrotactile and audio signal synthesis

```
// Code licensed under GNU GPL 3

(//execute here to store all synths
~tactileGain = 3;
~drumBuffer = Buffer.read(s, /*input directory/filename here*/);
```

```
~onsetBuffer = Buffer.alloc(s,512);
/*
SynthDef(\tactPreview, {|carfreq = 15,chan = 1,modfreq = 1,amp|
        var insig = SinOsc.ar(carfreq,mul:1)*SinOsc.ar(modfreq,mul: 0.5,add:0.5),
            outsig,env;
        outsig = SOS.ar(SOS.ar(insig,0.874225, -1.711427, 0.838289, 1.711427,
            -0.712514),
                0.980631, -1.922495, 0.941894, 1.922495, -0.922526);
        Out.ar(chan,outsig*amp);
}).add;
*/
/*
SynthDef(\drumloopPreview, {| out = 0,loop = 0,rate = 1,rfreq = 1000,chan1 = 0,chan2
     = 1 |
        var tactout, audout,bufnum;
        bufnum = ~drumBuffer;
        audout = PlayBuf.ar(1, bufnum, BufRateScale.kr(bufnum), doneAction:2,loop:
            loop,rate:rate);
        audout = RLPF.ar(audout,rfreq);
    Out.ar(0, audout);
}).add;
*/
//soundgen for prototype 2
SynthDef(\drumloop, {| out = 0,loop = 0,rate = 1,rfreq = 1000,chan1 = 0,chan2 = 1 |
        var tactout, audout,bufnum;
        bufnum = ~drumBuffer;
        audout = PlayBuf.ar(1, bufnum, BufRateScale.kr(bufnum), doneAction:2,loop:
            loop,rate:rate);
        audout = RLPF.ar(audout,rfreq);
        tactout = LPF.ar(SOS.ar(SOS.ar(audout,0.874225, -1.711427, 0.838289,
            1.711427, -0.712514),
                0.980631, -1.922495, 0.941894, 1.922495, -0.922526),800);
    Out.ar(0, [Out.ar(0,audout*0.5), Out.ar(1,tactout*~tactileGain)]);
}).add;

//soundgen for prototype 2 with onset detection
SynthDef(\drumloopOnsets,{
        |rfreq = 1000,modfreq=5,carfreq=100,rate = 1|
    var sig, chain, onsets, pips, synthsig,env,tactout,audout;

    sig = PlayBuf.ar(1, ~drumBuffer, BufRateScale.kr(~drumBuffer), loop: 1,rate:rate)
        ;
        audout = RLPF.ar(sig,rfreq);
```

```
    chain = FFT(~onsetBuffer, sig);
    onsets = Onsets.kr(chain, 0.7, \mkl);
        env = EnvGen.kr(Env.perc(0.01,0.2),gate: onsets);
        synthsig = SinOsc.ar(carfreq) * SinOsc.ar(modfreq,mul:0.5,add:0.5);
    synthsig = synthsig*env;
        tactout = SOS.ar(SOS.ar(synthsig,0.874225, -1.711427, 0.838289, 1.711427,
            -0.712514),
                0.980631, -1.922495, 0.941894, 1.922495, -0.922526);
    Out.ar(0, [Out.ar(0,audout*0.5), Out.ar(1,tactout*~tactileGain)]);
}).add;

// vibrotactile AM synth
SynthDef(\tactAM, {|carfreq,modfreq,amp,chan = 1|
        var insig = SinOsc.ar((carfreq.midicps)/2,mul:abs(amp))*SinOsc.ar(modfreq,mul
            : 0.5,add:0.5), outsig,env;
        env = EnvGen.kr(Env.perc(0.05,0.6),doneAction:2,levelScale:~tactileGain);
        outsig = LPF.ar(SOS.ar(SOS.ar(insig,0.874225, -1.711427, 0.838289, 1.711427,
            -0.712514),
                0.980631, -1.922495, 0.941894, 1.922495, -0.922526),800);
        Out.ar(chan,outsig*env);
}).add;

// vibrotactile sinusoidal synth
SynthDef(\tactSin, {|freq,amp,chan = 1|
        var insig = SinOsc.ar((freq.midicps)/2,mul:abs(amp)), outsig,env;
        env = EnvGen.kr(Env.perc(0.05,0.6),doneAction:2,levelScale:~tactileGain);
        outsig = SOS.ar(SOS.ar(insig,0.874225, -1.711427, 0.838289, 1.711427,
            -0.712514),
                        0.980631, -1.922495, 0.941894, 1.922495, -0.922526);
        Out.ar(chan,outsig*env);
}).add;

// used for boarder
SynthDef(\tactsineSynth,{
        |freq,chan = 1|
        var insig = SinOsc.ar(freq.midicps/2),outsig,env;
        env = EnvGen.kr(Env.perc(0.05,0.02),doneAction:2,levelScale:~tactileGain/1.5)
            ;
        outsig = SOS.ar(SOS.ar(insig,0.874225, -1.711427, 0.838289, 1.711427,
            -0.712514),
                        0.980631, -1.922495, 0.941894, 1.922495, -0.922526)*env;
        Out.ar(chan,outsig);
}).add;
```

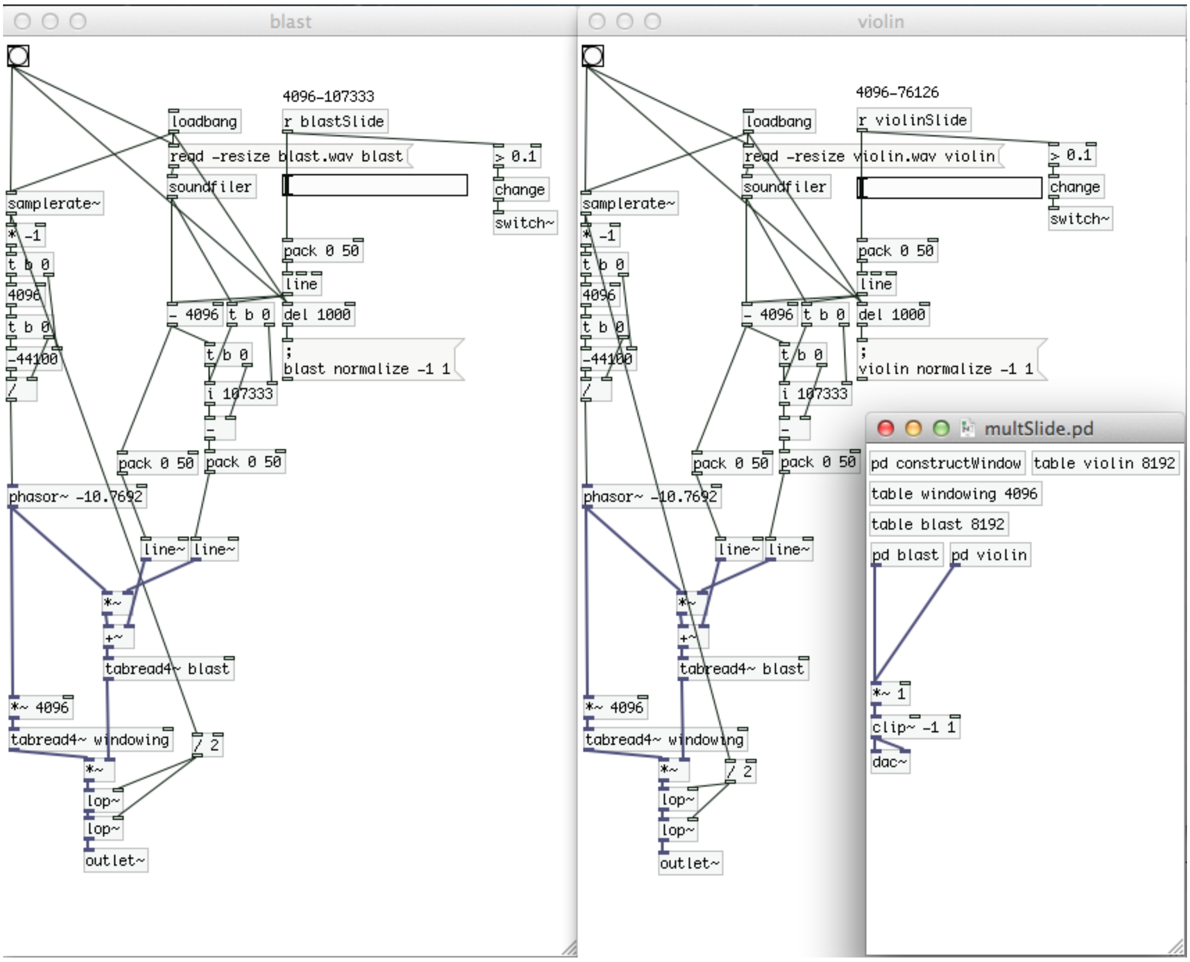
```
SynthDef(\tactImpSynthBoarder,{
        |amp = 1,chan = 1|
        Out.ar(chan,RLPF.ar(Impulse.ar(1)*EnvGen.ar(Env.new([1,0],[0.1,0.1]),
            doneAction:2,levelScale:50)*amp,250));
}).add;
// burst synthesis
SynthDef(\tactImpSynth,{
        |imppar,amp,time=1,freq,chan=1|
        var insig,outsig,env,env2;
        env2 = EnvGen.kr(Env.new([imppar,0],[time,time],\lin));
        insig = Impulse.ar(env2);
        env = EnvGen.kr(Env.new([0,1,1,0],[0,time,0.01],\exp),levelScale: 100,
            doneAction:2);
        insig = RLPF.ar(insig,freq.midicps,2);
        outsig = LPF.ar(SOS.ar(SOS.ar(insig,0.874225, -1.711427, 0.838289, 1.711427,
            -0.712514),
                0.980631, -1.922495, 0.941894, 1.922495, -0.922526),800);
        Out.ar(chan,outsig*env*abs(amp));
}).add;

// audio synth prototype 1. Tactsynth provided in right channel
SynthDef(\PMCrotaleMod, {
    |midi = 60, varpar = 1, art = 1,amp = 0.9,chan = 0,tactamp = 0|
    var env, out, mod, freq,tactout;
    freq = midi.midicps;
    env = Env.perc(0, art);
        mod = 5 + (1 / IRand(2, 6));
    out = PMOsc.ar(freq, mod*freq,
        pmindex: EnvGen.kr(env, timeScale: art, levelScale: varpar),
            mul: EnvGen.kr(env, timeScale: art, levelScale: 0.5));
    out = out * EnvGen.kr(env, timeScale: 1.3 * art,
        levelScale: Rand(0.1, 0.5), doneAction: 2);
        tactout = LPF.ar(SOS.ar(SOS.ar(out,0.874225, -1.711427, 0.838289, 1.711427,
            -0.712514),
                0.980631, -1.922495, 0.941894, 1.922495, -0.922526),800);
        Out.ar(chan, Pan2.ar(out*abs(amp),-1,1) + Pan2.ar(tactout,1,tactamp));
}).add;
)
```

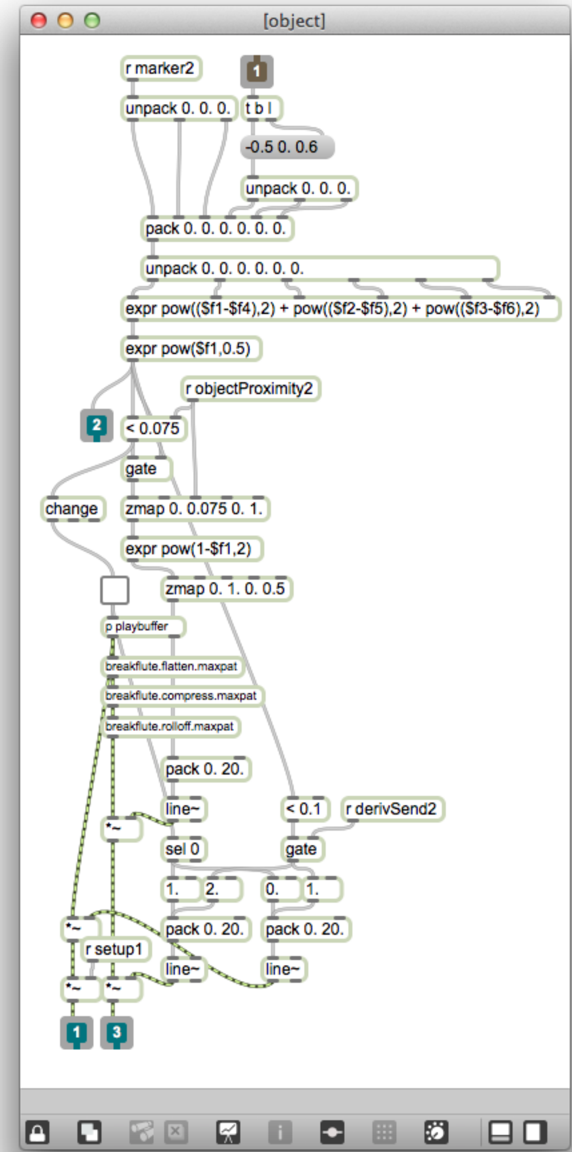
Appendix B

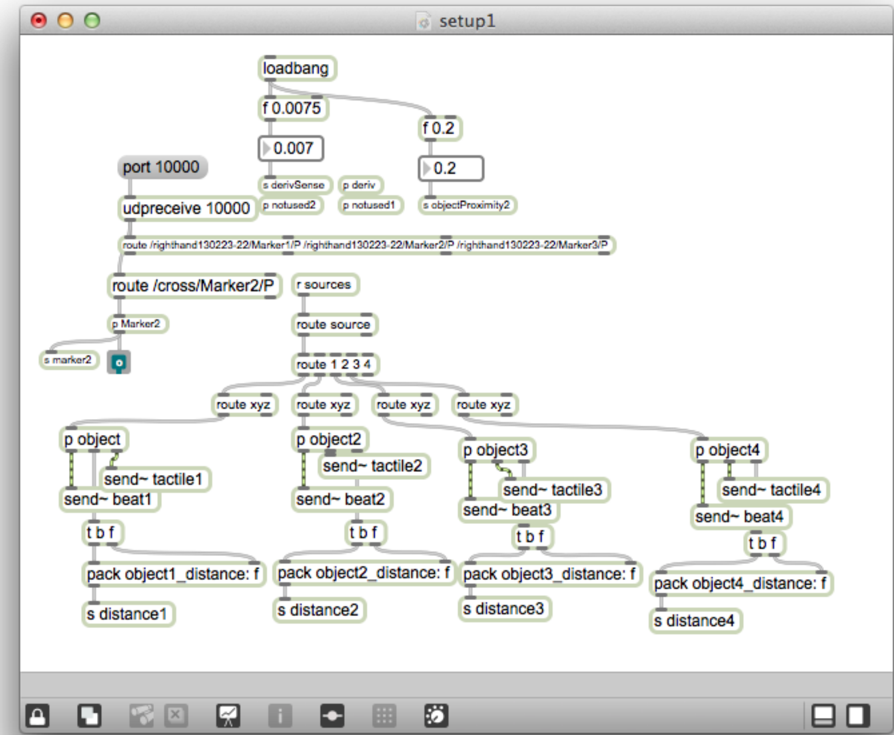
Pd and Max MSP Patches

B.1 Pd Patch for Android Phone



B.2 Max MSP Patches for Exploration




setup1
loadbang
f 0.0075
f 0.2
0.007
0.2
port 10000
udpreceive 10000
s derivSense
p deriv
p notused2
p notused1
s objectProximity2
route /righthand130223-22/Marker1/P /righthand130223-22/Marker2/P /righthand130223-22/Marker3/P
route /cross/Marker2/P
r sources
p Marker2
route source
s marker2
route 1 2 3 4
route xyz
route xyz
route xyz
route xyz
p object
p object2
p object3
p object4
send~ tactile1
send~ tactile2
send~ tactile3
send~ tactile4
send~ beat1
send~ beat2
send~ beat3
send~ beat4
t b f
t b f
t b f
t b f
pack object1_distance: f
pack object2_distance: f
pack object3_distance: f
pack object4_distance: f
s distance1
s distance2
s distance3
s distance4

Appendix C

Matlab Code

C.1 Code for Figure 2.4 and Figure 2.5

```
% This script creates the plots of unipolar and bipolar
% amplitude modulated signals as shown in chapter 2 in the thesis
% Licensed under GNU GPL v3
t = 0:(1/1000):1; t = t(1:length(t)-1);
bmod = cos(t*4*pi);
umod = .5.*cos(t*4*pi); umod = (ones(1,length(umod)).*0.5)+umod;
bmod2 = cos(t*40*pi);
umod2 = .5.*cos(t*40*pi); umod2 = (ones(1,length(umod2)).*0.5)+umod2;
car = cos(t*2*pi*250);
%wmod = window(@blackman,500);

umodded = car.*umod; bmodded = car.*bmod; umodded2 = car.*umod2;
bmodded2 = car.*bmod2; U = fft(umodded,1000); B = fft(bmodded,1000);
U2 = fft(umodded2,1000); B2 = fft(bmodded2,1000);
Urange = 0:((length(U)-1)/2);

figure(1);
subplot(2,2,1); plot(t,umodded); xlabel('Time␣(s.)');
ylabel('Amplitude'); title('Unipolar␣amplitude␣modulation');

subplot(2,2,2); plot(t,bmodded); xlabel('Time␣(s.)');
ylabel('Amplitude'); title('Bipolar␣amplitude␣modulation');

subplot(2,2,3);
plot(0:((length(U)-1)/2),abs(U(1:(length(U)/2)))/max(abs(U(1:(length(U)/2)))));
set(gca,'XLim',[240 260]); xlabel('Frequency␣(Hz)');
ylabel('Normalized␣Magnitude');
```

```
title('Spectrum␣of␣unipolar␣amplitude␣modulation');

subplot(2,2,4);
plot(0:((length(B)-1)/2),abs(B(1:(length(U)/2)))/max(abs(B(1:(length(U)/2)))));
set(gca,'XLim',[240 260]);
title('Spectrum␣of␣bipolar␣amplitude␣modulation');
xlabel('Frequency␣(Hz)');
ylabel('Normalized␣Magnitude');

%%%%%%%
figure(2);
subplot(2,2,1);
plot(t,umodded2);
xlabel('Time␣(s.)');
ylabel('Amplitude');
title('Unipolar␣amplitude␣modulation');
%title('f_{carrier} = 250 Hz, f_{modulation} = 20 Hz, unipolar');

subplot(2,2,2);
plot(t,bmodded2);
xlabel('Time␣(s.)');
ylabel('Amplitude');
title('Bipolar␣amplitude␣modulation');

subplot(2,2,3);
plot(0:((length(U2)-1)/2),abs(U2(1:(length(U2)/2)))/max(abs(U2(1:(length(U2)/2)))));
set(gca,'XLim',[220 280]);
xlabel('Frequency␣(Hz)');
ylabel('Normalized␣Magnitude');
title('Spectrum␣of␣unipolar␣amplitude␣modulation');

subplot(2,2,4);
plot(0:((length(B2)-1)/2),abs(B2(1:(length(U2)/2)))/max(abs(B2(1:(length(U2)/2)))));
set(gca,'XLim',[220 280]);
%title('Spectrum of signal modulated by bipolar signal');
title('Spectrum␣of␣bipolar␣amplitude␣modulation');
xlabel('Frequency␣(Hz)');
ylabel('Normalized␣Magnitude');
```

C.2 Generating sinesweeps

```
% sinesweeps are generated using code based on Berdahl:

function sdbl = generate_sinesweeps(f1,f2,fs,N)
% sdbl = generate_sinesweeps(f1,f2,fs,N)
%
% f1: starting frequency [Hz]
% f2: ending frequency [Hz]
% fs: sampling rate (Hz)
% N: If N is an integer, the total length of the excitation sound file is
% 2*(2^N) samples, or 2*(2^N)/fs seconds. This helps speed up
% computation.
%
% A sine sweep ranging from f1 to f2 is created. It is repeated
% twice so that cyclical (de)convolution may be applied to easily
% find the inverse filter. The result is also written to 'sinesweeps.wav'
%
% T: the length of the excitation in samples
%
%
% RealSimPLE Project
% Edgar Berdahl, 6/10/07
% Updated on 8/19/08
%
% e.g. generate_sinesweeps(20,20000,44100,17);



T = (2^N)/fs;

% Create the swept sine tone
w1 = 2*pi*f1;
w2 = 2*pi*f2;
K = T*w1/log(w2/w1);
L = T/log(w2/w1);
t = linspace(0,T-1/fs,fs*T);
s = sin(K*(exp(t/L) - 1));
%impsync = zeros(1,200);
%impsync(100) = 1;



% Double the length so that it is easy to use cyclical (de)convolution
sdbl = [s zeros(1,200) s zeros(1,200)];
```

```
%window = window(@hamming,length(s));

% Scaling by 0.9999 suppresses a warning message about clipping.
wavwrite(sdbl*0.8,fs,sprintf('Sinesweeps_%i_%i_%i.wav',f1,f2,fs));
sdbl = 0.8 * sdbl;
```

C.3 Obtaining Response

```
% code written by Marcello Giordano
% declares a function that can be used to
% output and record sinesweeps with the
% national instruments acquisition board
% Licensed under GNU GPL 3


function [in,out,tmp] = dev1_setup(f1,f2,fs,sweep,take)

%ai=analoginput('ni','Dev1');
%addchannel(ai,0);
%ao=analogoutput('ni','Dev1');
%addchannel(ao,0);

myDaq = daq.createSession('ni');
myDaq.addAnalogOutputChannel('Dev1', 'ao0', 'Voltage');
myDaq.addAnalogInputChannel('Dev1', 'ai0', 'Voltage');
myDaq.addAnalogInputChannel('Dev1', 'ai1', 'Voltage');
myDaq.Rate = fs;
in = zeros(length(sweep),4);
out = zeros(length(sweep),4);
tmp = zeros(length(sweep),2);
for i = 1:4
    myDaq.queueOutputData(sweep');
    myDaq.wait(1000);
    tmp = myDaq.startForeground;
    in(:,i) = tmp(:,2);
    out(:,i) = tmp(:,1);
    %r(:,i) = myDaq.startForeground;
end
%m = zeros(length(sweep));
%for i = 1:length(sweep)
%m = mean(r',2);
```

```
%data = m';
%myDaq.queueOutputData(sweep');
%myDaq.wait(1000);
%data = myDaq.startForeground;
%audiowrite(sprintf('Resp_take%i_%i_%i_%i.wav',take,f1,f2,fs),data,fs);
csvwrite(sprintf('Input_take%i_%i_%i_%i.txt',take,f1,f2,fs),in);
csvwrite(sprintf('Output_take%i_%i_%i_%i.txt',take,f1,f2,fs),out);
myDaq.release;
```

C.4 Analysis of Sparkfun Amplifier

```
% This script is used to obtain the frequency response of the
% sparkfun class d mono audio amplifier using the recorded input and
% output of the sinesweep
% Licensed under GNU GPL v3

aa = csvread('Input_take1_1_1100_5000.txt');
bb = csvread('Output_take1_1_1100_5000.txt');
cc = csvread('Input_take1_1_2500_8000.txt');
dd = csvread('Output_take1_1_2500_8000.txt');

hlfi = aa(65737:131472,1);
hlfo = bb(65737:131472,1);
hlfil = cc(65737:131472,1);
hlfol = dd(65737:131472,1);
spfnw = window(@blackman, 65736);

% perform
DDR = fft(hlfo.*spfnw)./fft(hlfi.*spfnw);
DDRU = fft(hlfo)./fft(hlfi);
DDRL = fft(hlfol.*spfnw)./fft(hlfil.*spfnw);
DDRLU = fft(hlfol)./fft(hlfil);
range = 0:32867; range = range'; range = range./32867.*2500;
range2 = 0:32867; range2 = range2'; range2 = range2./32867.*4000;

% create matrices containing the dB magnitude response
cropped = 20*log10(abs(DDR(1:32868)));
cropped2 = 20*log10(abs(DDRU(1:32868)));
cropped3 = 20*log10(abs(DDRL(1:32868)));
cropped4 = 20*log10(abs(DDRLU(1:32868)));

```

```
%% this plot was used in the thesis. The window is rectangular
figure(1);
plot(range2,cropped4);
title('Magnitude␣response␣of␣Sparkfun␣amplifier');
set(gca,'XLim',[0 1000]);
xlabel('Frequency␣(Hz)');
ylabel('Decibel␣(dB)');
set(gca,'YLim',[2 6.5]);

% figure(1);
% plot(range,cropped);
% title('Magnitude response of Sparkfun amplifier (Blackman window)');
% set(gca,'XLim',[0 1000]);
% xlabel('Frequency (Hz)');
% ylabel('Decibel (dB)');
% set(gca,'YLim',[-10 7]);
%
% figure(2);
% plot(range,cropped2);
% title('Magnitude response of Sparkfun amplifier (Rectangular window)');
% set(gca,'XLim',[0 1000]);
% xlabel('Frequency (Hz)');
% ylabel('Decibel (dB)');
% set(gca,'YLim',[-10 7]);
%
% figure(3);
% plot(range2,cropped3);
% title('Magnitude response of Sparkfun amplifier (Blackman window)');
% set(gca,'XLim',[0 1000]);
% xlabel('Frequency (Hz)');
% ylabel('Decibel (dB)');
% set(gca,'YLim',[-10 7]);
```

C.5 Code for Figure 3.11 and Figure 3.14

```
% This script creates the plot of the author's interaction with
% prototype 1 as shown in Chapter 3
% Licensed under GNU GPL 3

% Importing recorded marker position data:
tactAmx = importdata('markerX'); tactAmy = importdata('markerY');
```

```
tactAmz = importdata('markerZ'); tactAmvel = importdata('markerYVel');
% creating a time representation vector
tRange = (0:length(tactAmx)-1)./length(tactAmx)*20;
tRange2 = (0:length(tactAmx)-1)./length(tactAmx)*279+150;

% Audio and the tactile signals are imported using
% the built in function provided by the mirtoolbox:
tactAm = miraudio('tactAm.aiff');
audAm = miraudio('tactAmSound_6db_boost.aif');
% Computing spectrograms
sTactAm = mirgetdata(mirspectrum(tactAm,'Frame',4096/44100,.5,'Max',400,'Length',16000));
sAudAm = mirgetdata(mirspectrum(audAm,'Frame',4096/44100,.5,'Max',750,'Length',16000));

% creating figure and enabling greyscale
hFig = figure(1); a = gray(100); a = 1-a; colormap(a);
set(hFig, 'Position', [1.0 1.0 1243.0 650.0]);

% Plotting
subtightplot(3,1,1,0);
aa1 = [0:100:400]; aa2 = aa1*(147/400)+0.5;
blanky = zeros(1,length(tRange2));
imagesc(tRange,0:400,sTactAm); axis xy;
set(gca,'XLim',[7 20]);
set(gca,'XTick',[7:20]);
set(gca,'XTickLabel',[]);
ylabel('Frequency␣(Hz)','FontSize',16);
ylhand = get(gca,'ylabel');
set(ylhand,'FontSize',16);
set(gca,'YTickLabel',aa1,'FontSize',14);
text(8.59,263,'\leftarrow␣Note␣selection␣feedback','FontSize',16);
title('Spectrogram␣of␣vibrotactile␣signal␣(top),␣audio␣signal␣(middle)␣and␣marker␣x,y,z␣position␣(lower)','FontSize',16);

subtightplot(3,1,2,0);
bb1 = [0:100:650]; bb2 = bb1*(274/750)+0.5;
imagesc(tRange,[0:100:750],sAudAm); axis xy;
set(gca,'YTick',bb1);
set(gca,'YTickLabel',bb1,'FontSize',14);
set(gca,'XLim',[7 20]);
set(gca,'XTick',[7:20]);
set(gca,'XTickLabel',[]);
```

```
ylabel('Frequency␣(Hz)','FontSize',16);

subtightplot(3,1,3,0);
plot(tRange,tactAmx,'k-',tRange,tactAmy,'k--',tRange,tactAmz,'k:');
set(gca,'XLim',[7 20]);
set(gca,'XTick',[7:20]);
set(gca,'XTickLabel',[0:13]);
set(gca,'YLim',[-500 2200]);
set(gca,'YTick',[-500:500:2200]);
set(gca,'YTickLabel',[-500:500:2200],'FontSize',14);
xlabel('Time␣(s)','FontSize',16);
xlabh = get(gca,'XLabel');
set(xlabh,'Position',get(xlabh,'Position') + [0 200. 0]);
ylabel('Marker␣pos␣(mm)','FontSize',16);
h_legend = legend('x-pos','y-pos','z-pos');
set(h_legend, 'Location','NorthWest','FontSize',16);
```

```
% This script creates the plot of the author's interaction with
% prototype 2 as shown in Chapter 3
% Licensed under GNU GPL 3

% Importing recorded marker position data:
tactAmx = importdata('markerX'); tactAmy = importdata('markerY');
tactAmz = importdata('markerZ'); tactAmvel = importdata('markerYVel');
% creating a time representation vector
tRange = (0:length(tactAmx)-1)./length(tactAmx)*20;

% Audio and the tactile signals are imported using
% the built in function provided by the mirtoolbox:
tactAm = miraudio('drumtact.aiff','Normal');
audAm = miraudio('drumaud.aiff','Normal');

% Computing spectrograms
sTactAm = mirgetdata(mirspectrum(tactAm,'Frame',4096/44100,.5,'Max',400,'Length',16000));
sAudAm = mirgetdata(mirspectrum(audAm,'Frame',4096/44100,.5,'Max',1500,'Length',16000));

% creating figure and enabling greyscale
hFig = figure(1); a = gray(100); a = 1-a; colormap(a);
set(hFig, 'Position', [1.0 1.0 1243.0 650.0]);

% Plotting:
```

```
subplot(3,1,1);
subtightplot(3,1,1,0);
imagesc(tRange,0:400,sTactAm); axis xy;
aa1 = [0:100:400]; aa2 = aa1*(147/400)+0.5;
title('Spectrogram␣of␣vibrotactile␣signal␣(top),␣audio␣signal␣(middle)␣and␣marker␣x,y,z␣position␣(lower)','FontSize',16);
set(gca,'XLim',[7 20]);
set(gca,'XTick',[7:20]);
set(gca,'XTickLabel',[]);
ylabel('Frequency␣(Hz)','FontSize',16);
ylhand = get(gca,'ylabel');
set(ylhand,'FontSize',16);
set(gca,'YTickLabel',aa1,'FontSize',14);

subplot(3,1,2);
subtightplot(3,1,2,0);
bb1 = [0:250:1250];
imagesc(tRange,0:1500,sAudAm); axis xy;
set(gca,'XLim',[7 20]);
set(gca,'XTick',[7:20]);
set(gca,'XTickLabel',[]);
set(gca,'YTick',bb1);
set(gca,'YTickLabel',bb1,'FontSize',14);
ylabel('Frequency␣(Hz)','FontSize',16);
ylhand = get(gca,'ylabel');
set(ylhand,'FontSize',16);

subtightplot(3,1,3,0);
plot(tRange,tactAmx,'k-',tRange,tactAmy,'k--',tRange,tactAmz,'k:');
set(gca,'XLim',[7 20]);
set(gca,'XTick',[7:20]);
set(gca,'XTickLabel',[0:13]);
set(gca,'YLim',[-1500 2200]);
set(gca,'YTick',[-1500:500:2000]);
set(gca,'YTickLabel',[-1500:500:2000],'FontSize',14);
xlabel('Time␣(s)','FontSize',16);
xlabh = get(gca,'XLabel');
set(xlabh,'Position',get(xlabh,'Position') + [0 200. 0]);
ylabel('Marker␣pos␣(mm)','FontSize',16);
h_legend = legend('x-pos','y-pos','z-pos');
set(h_legend, 'Location','NorthWest','FontSize',16);
```