An evaluation of the interstitial beat across the modalities of touch and sound for the characterization of a meaningful haptic metronome

Nick Pourazima

July 30, 2018

Music and Technology
College of Fine Arts
Carnegie Mellon University
Pittsburgh, PA

Thesis Committee:
Professor Stephen Neely
Professor Thomas Sullivan
Professor Jesse Stiles

Submitted in partial fulfillment of the requirements for the degree of Masters of Science in Music and Technology.

Keywords: sensorimotor synchronization, vibrotactile perception, haptic, metronome, modality, entrainment

To my grandmother, for early guidance towards a lifelong quest of knowledge and truth.

Abstract

The following work is an expansion of sensorimotor synchronization research. It provides an evaluation of the intervening space between the beat as it applies to the modalities of touch and sound. The crux of the experiment is a tap test comparison of continuous and discrete impulses over static (isochronous) and dynamic (non-isochronous) pulse intervals.

Time based response metrics of a wearable haptic are contrasted to a suite of audible tests. Though vast evidence promotes an auditory advantage in guiding rhythmic accuracy and low asynchrony, this work hypothesizes a haptic benefit when the dynamically changing beat is occupied with a continuous wave across the modality of touch.

The analysis of 16 subjects (8 professionals, 8 amateur and non-musicians) resulted in favorable results for the haptic device during the dynamic test cases as contrasted to the auditory test results. Though the auditory modality yielded the best results for the isochronous test cases, the haptic device won out for non-isochronous or dynamically changing beats with a much cleaner standard deviation. This implies a greater synchronization ability with the haptic device and strongly supports the hypothesis of this work.

The overarching goal is to inform validity and design of a haptic wearable which seeks to supplant the traditional metronome experience in providing a meaningful gestural system. The work holds value towards future entrainment studies in expressive musical performance but can be expanded to include extra-musical applications such as stroke and Parkinson's patient gait rehabilitation practice.

Acknowledgments

First and foremost to whom ideation credit is due, my advisor Professor Stephen Neely. Our bi-weekly discussions kept me on the right path. Thank you for the guidance and experience you brought to this project. I hope this proves to be exemplary to your design research as well as the framework for future work to come.

To Professor Sullivan and Stiles for their accessibility, open-mindedness, and overall support from both a creative and technical standpoint.

To my roommates and close friends, Mike and Craig, for those long nights of brainstorming possibilities and debugging issues. Thank you for not only being my think tank but for keeping me inspired and grounded. Also my colleague Garrett, for the drive, motivation, and support in proofreading this work.

Last but certainly not least, a special thank you to Rachel for the unending love and support in keeping the light at the end of the tunnel shining.

Funding

This work was supported by Professor Stephen Neely.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The interstice is an intervening space. When applied to a rhythmic context, the interstitial beat can be represented by two distinct states; whether energy exists within this small moment in time or if it does not.

The underlying question, as applied to the daily practice of a trained musician or the innate entrainment[1] of the average human being, is whether the space between the beat matters. Does filling the space provide an added awareness or preparation for upcoming onsets?

The objective of this work is to quantitatively discern whether a continuous wave, one which leads up to the maximum amplitude of the beat and trails off into a smooth decay, exhibits differentiation from it's instantaneous counterpart in communicating regular or irregular pulses. To quantify this differentiation, an expansive set of analog and discrete tap synchronization tests spanning the modalities of sound and touch are conducted across a group of musicians, amateurs, and non-musicians.

Ancillary to this work, a haptic wearable is prototyped and evaluated for design optimization with an overarching goal of translating the gestural motion of the conductor.

The project presents a unique opportunity to enable expansion of the existing sensorimotor synchronization findings to the haptic modality in continuous form, with the intent to resolve the inquiry as to whether filling in the space between the beat, the interstitial, has a positive impact in communicating dynamic changes more

[1] Within this context entrainment will refer to external rhythmic synchronization.

effectively.

1.1 Motivation

While it is clear that nearly every professional musician has honed technique over countless hours of practice to an audible metronome, it is not directly obvious whether they have ingrained a true sense of rhythm at the foundational level with the primary instrument of expression, the body itself.

Intrinsic awareness to subtle nuances of tempo remains a subject commonly unexposed to a student in training. Yet this ability, to perform in the spaces surrounding the beat, defines the difference between a rigid performance and one that flows with elasticity and musical expression.

Is there missing information from the daily practice of a trained musician to an audible metronome? In a traditional sense, the audible click of the metronome minimizes the interstitial space with an instantaneous (or discrete) impulse signal. The pendulum motion exhibited also seeks to convey meaningful rhythmic information with the space it occupies in the visual modality, much like the gestural motion of a conductor. Although an excellent tool in establishing a sense of musical time and precision, the danger in use of such a mechanical object lies within the mathematical exactitude according to American composer and music critic Daniel Gregory Mason. Therefore manifesting a lifelessness where instead a living and breathing musical entity should exist with its own "ebb and flow of rhythmical energy"[1].

1.2 Background

Humans have an innate ability to not only notice periodic movement, but to mimic and adapt to changes in their environment [2]. The extent of these capacities can be expanded through training which consists of recognizing, retaining, analyzing, and reproducing rhythms [3]. The results of which can hold extra-musical impact as rhythmic stability finds important applications in everyday life.

Dalcroze Eurythmics is a curriculum developed by composer and educator Emile Jaques-Dalcroze in the early 1900's that has sought to integrate natural musical expression via movement [4]. Through a series of exercises the instructor ushers his students to coordinate movement to varying levels of rhythmic push and pull. A student might be conducting a subdivision heard in the melody with his/her hands while simultaneously stepping in coordination to the fundamental pulse played in the harmony. A sense of constant forward motion pervades the actions of the student. This overall embodiment of continuity seeks to permeate all elements of musicality development.

From nearly two decades of work as a licensed Dalcroze instructor at Carnegie Mellon University, Professor Stephen Neely has implemented these techniques. His current research in design seeks to further explore the interstitial. In doing so he imparts an inquiry as to what is gained in attempting to fill the space between the *crusis*[2] with a natural analog wave, one that provides a build up and decay common to natural happenings, much like the gestural motion of a conductor.

1.3 Sensorimotor synchronization

Research surrounding the psychology of rhythmic perception is grounded within the framework of *sensorimotor synchronization* (*SMS*) or the coordination of rhythmic movement to an external rhythm.

What follows is a brief overview of SMS. Critical terminology is defined in 1.3.1, followed by a primer of available tap test software in 1.3.2. A framework for the current work is established through discoveries illuminated by prior SMS research in 1.3.3. Finally, a brief discussion of both biomechanical and perception limitations in 1.3.4 concludes this chapter.

1.3.1 Terminology

The main method of data collection for SMS tap based tests involve calculation of the time delta between the tap and event onset, called the *asynchrony*. Within the context of this work it shall be defined as:

$$TapOnset - TrueOnset$$

[2] The click moment of the beat.

The mean of this asynchrony is typically negative (*NMA*), indicative of the participants anticipation of the beat rather than reaction. Positive asynchronies imply a reactive approach thereby unfavorable within the context of this work. The standard deviation of the asynchrony, SD_{asy}, is an index of stability; lower values indicative of a better level of synchronization [5]. For the purposes of this research, SD_{asy} is predominantly shown superimposed onto bar charts.

Other important metrics include the variability and mean of the inter tap interval (ITI) and the inter onset interval (IOI), or the time between successive beats, measured in milliseconds. Mismatch between the ITI and IOI implies poor synchronization skill from the participant.

Phase Correction Response is defined as the shift of the immediately following tap from its expected time point, given by:

$$PCR = (TapOnset_{n+1} - TrueOnset_{n+1}) - (TapOnset_n - TrueOnset_n)$$

When a participant is instructed to tap on the beat, this is termed 1:1 synchronization. 4:1 synchronization, for example, is a beat subdivided into four with one tap on the beat. Subdivision tests typically yield lower mean (SD_{asy}) values [5]. This work will focus on 1:1 synchronization as discussed in 4.1.1.

1.3.2 Tap Test Software

The finger tap mechanism holds strongest precedence in SMS research due to its reliability, precision *(ms)*, and discrete nature. Studies predominantly rely on a MIDI based (drum pad) instrument to register tap events and provide some sort of auditory feedback. A few tap based software suites for experimentation and data acquisition are readily available: a Linux based system written by Finney in 2001 named *FTAP* [6], and a *Matlab* based toolbox by Elliot in 2009 called *MatTAP* [7]. FTAP relies on a MIDI source with a reported mean auditory latency of approximately 14.6 ms (SD = 2.8) [8]. Superfluous and unregistered taps were common.

Both *FTAP* and *MatTap* were viable options for this work but ultimately deemed either outmoded, lacking multi-threaded and high baudrate (115200) hardware support for haptic integration over serial, or incompatible with the system architecture in use[3].

In a novel high-precision, low-latency approach by Prof. Schultz in 2015 at the University of Montreal [8], an Arduino force sensitive resistor (FSR) based tap mechanism was constructed. Latency between time of tap and auditory feedback was minimized with a mean of 0.6 ms (SD = 0.3). The results also demonstrated the reliability of the FSR in recording fewer superfluous taps as well as fewer missed taps.

It was inevitably decided to construct custom hardware and software to fit the test needs of the project as discussed in Section B.1 with a latency breakdown discussed in B.3.

1.3.3 Expectations

This section focusses on key insights with respect to the auditory domain from prior SMS research which inform expectations for the data analysis conducted in Chapter 4.

Variability

The asynchrony variability (SD_{asy}) is generally lower in professional musicians than non-musicians or those with no prior musical exposure. In an isochronous test with an IOI of 500 ms (120 bpm), no difference in SD_{asy} was found between amateurs and non-musicians [5]. The data analysis of this work will therefore be grouped into professionals versus non-musicians and amateurs.

Percussionists and pianists had the lowest asynchronies of all musicians. This might imply that a high level of rhythmic expertise reduces variability of tapping but due to the percussive nature of the instruments becomes hard to determine. Furthermore, as the duration of the IOI increases, SD_{asy} increased in a non linear fashion.

When professionals migrated away from the tap test and instead used their native instruments, the results were greatly improved. It was reported by Stoklasa, Liebermann, and Fischinger in a paper presented at the Music Perception and Cognition in 2012 that musicians playing their own brass or string instrument in

[3] Macbook Pro Retina Mid-2012 OSX 2.6GHz Intel I7 10.11.6.

synchrony with a metronome showed an asynchrony of 2 ms, unlike their tapping results of −13 ms [5].

Negative Mean Asynchrony

The NMA is typically smaller for musicians and remains relatively constant throughout a changing IOI. A linear increase in NMA as the IOI increases can be expected from non-musicians [5]. As expected, nonisochronous tapping introduces distortions within the ITI as opposed to steady or isochronous patterns. This had a tendency to affect local asynchronies but the global NMA remained persistent [9].

As a counter to the proposed hypothesis, Bialunkska et al. argued that the reason for a negative mean asynchrony was due to faster sensory accumulation from the auditory and visual modalities than from tactile feedback received from taps [10]. The expectation was thus a dependence of the sensory accumulation rate to the stimulus intensity. This was later found to be uncorrelated.

A positive mean asynchrony is expected as the IOI approaches the biomechanical limit of execution as discovered by Krause, Pollok, and Schnitzler in 2010 [11].

Auditory Dominance in SMS

SMS research historically identifies what is known as an auditory advantage, or the dominance of the auditory motor connection within the task of beat synchronization. The auditory advantage is discussed in detail in Section 2.1. Recent studies have proven that given meaningful spatiotemporal information, as in the bouncing ball example discussed in 2.4, synchronization to alternate modalities can almost be as good as to an auditory metronome.

1.3.4 Rate Limits

In order to impose valid constraints on the tests carried out in this work, it is important to understand the SMS rate limits.

According to experiments done by Keele, Pokorny, Corcos, and Ivry in 1985, the calculation for the fastest absolute response time possible for a tap based test can be divided into perception and the biomechanical limit [12]:

1. Biomechanical Limit[4]:
    - Between the 5 - 7 Hz range, or a period of 150 - 200 ms [13].
2. When discussing a perceptual basis, SMS tests are valid within a particular temporal range:
    - For audio based tests with 4:1 synchronization:
        - The upper rate limit was shown to be as high as 8 - 10 Hz, or a period of 100 - 125 ms (approximately 600 bpm).
        - The lower rate limit was modality independent and found to be 0.56 Hz, or a period of 1800 ms (33 bpm) [13].
    - Visual stimuli was found to be less than 2.5 Hz, or a period of 400 ms (150 bpm).
    - The haptic design section A.1.1 discusses the rate limits of touch.

In order to establish a middle ground and determine a fair and effective IOI window across musicians and non-musicians, this work has chosen to focus on tempi ranging from **45 - 180 bpm**, or a period of **333 - 1333 ms**. The selection presents an opportunity to test both the higher and lower bounds of relative ability and noise for the haptic metronome past the biomechanical limit range (> 150 bpm).

[4]Defined as the fastest possible motor speed of a tapping action.

Chapter 2

Prior Work

This chapter begins with an overview of previous SMS research in an attempt to debunk what is known as the auditory advantage 2.1. Three specific research projects are then discussed, each directly informing design of the haptic built for this project: the haptic drum kit 2.2, vibrotactile metronome 2.3, and the continuous visual metronome 2.4. To conclude, 2.4.1 grants a brief insight towards commercially realized products relevant to the field.

2.1 Auditory Advantage

Most SMS research leans toward the notion of a distinct advantage from the discretely timed auditory stimulus, implying that the neural and evolutionary mechanisms underlying beat synchronization are modality-specific. The following examples are proponents of the counterargument, predominantly involving the relatively new field of touch.

2.1.1 Multisensory Cues

Maintaining synchrony with a periodic event requires that the central nervous system (CNS) compensate for timing variation arising from sensory, decision and motor processing noise. Keeping in time with a pacing source (metronome) requires continual corrections based on the timing error (asynchrony) between the metronome and performed actions. The CNS can alternate between cues depending on the demand of the task or combine info from different senses. In the context of rhythmic cues the brain will weigh signals according to the relative reliability in the timing of the events across modalities, ensuring optimal movement production to the underlying event extracted from the signals.

In an experiment carried out by Elliott in 2010, it was discovered that the variability of asynchrony for unimodal tactile cues was lower than for the visual metronome ($F1,9 = 6.929$, $P = 0.027$) and only slightly higher than that for unimodal auditory cues [14].

2.1.2 The Tactile Modality

A 2016 study by the Department of Psychology at Ryerson University considered whether the auditory advantage persisted across the tactile modality [15]. The experiment was a tap test of non musicians put through a series of simple and complex rhythmic sequences with a varied area of haptic stimulation. In conditions involving a large area of stimulation and simple rhythmic sequences, tactile synchronization closely matched auditory. They proved that if made salient enough, the accuracy of synchronization to a tactile metronome can equal synchronization to an auditory metronome, further challenging the idea of an auditory advantage over all other modalities for synchronization to discretely timed rhythmic stimuli. However, auditory won out for synchronization of complex rhythmic sequences.

2.2 Haptic Drumkit

In 2010, a group at the Open University designed a haptic which would enable a drummer to learn multi-limb coordination with the broader goal of polyrhythmic entrainment. They too adopted Dalcroze entrainment theories for guiding rhythmic perception.

Four haptics were worn on the wrists and ankles of five participants. Each haptic consisted of a single vibrotactile and as such operated in an instantaneous mode. The devices would communicate in synchronization to a singular beat, while an individual device could vibrate to the subdivision. Each stimulation represented an action which the drummer would take [3].

In a user survey following the tests, it was mentioned that the haptic guidance felt intimate. The users appreciated not having to work out the division of labor like one would with the audible modality. However, the drummming had a tendency to drown out the vibrotactile signal and the attack was seemingly blurred or lost to the noise as the haptics reached higher tempo.

It was found that a haptic had to be on for a minimum of 50 ms in order to be felt completely. Furthermore, a minimum gap of 50 ms would have to be in between each pulse in order for two pulses to feel distinct. Another frequent request was to reposition to just the arms or to have the option to disable the ankles as it was found to be very difficult to feel on the lower limbs.

This research gave insight into training with haptic stimuli alone, and is a strong proponent for utilization of the touch modality within the context of Dalcroze training. Throughout their research an important distinction was made between stimulus response and fostering entrainment. It was found that a more optimal solution in the future would promote more of a proactive rather than reactionary response. Further, replacement of the vibrotactile with a tactor (see Appendix A.1.2) in the next revision promoted a cleaner signal with wider dynamic range and finer temporal resolution.

2.3 Vibrotactile Metronome

The *Vibrotactile Metronome* is the current thesis project of Patrick Ignoto of the Centre for Interdisciplinary Research in Music Media and Technology (CIRMMT) program at McGill University.

The work has some fascinating parallels to this project and has granted some very tangible insights towards testing and overall procedure. Patricks overall goal was to propose a device which uses tactile sense to provide similar functionality to a click track as its used for a contemporary classical music conductor with the added benefit of not blocking the ear or interfering with the conductors perception [16].

The haptic design consisted of two ERMs and one pager buzzer. A transmitter was connected to a PC running the visual programming language Max. Real-time audio analysis of the incoming signal was completed using the bonk object to find downbeats. This triggered the generation of the vibrotactile envelope signal and a control message was transmitted to the device. The pulses were to feel continuous and not discrete, mimicking that of a pendulum motion. The pulses peak amplitude had to line up with the audio track. To synchronize with the audio click he triggered the haptic pulse midway between two audio clicks.

At the end of the proposal, the real-time setup was migrated to post-processing in Matlab since the hardware was having trouble in keeping up with the buffering. Not much insight was given as to how the vibrotactile waveform was shaped; it is assumed that some level of motor control was involved.

2.4 A Continuous Visual Metronome

In a novel advancement challenging the auditory advantage and perhaps paving the way towards a more meaningful gesture, researchers in the Psychology department at Sun Yat-Sin University in Guangdong found continuous motion of a bouncing ball to be as stable as synchronization to an auditory metronome [17].

Stability of beat synchronization to discrete visual modalities, such as a flash of light, had been previously shown to be less stable that its auditory counterpart. However, the design and impact of a continuous visual metronome had yet to be explored. The team designed a visual bouncing ball simulation which manipulated acceleration animation to fit the IOI desired. Participants were instructed to tap to the bottom position of the bounce. Results for the bouncing ball visual were nearly equivalent to the auditory control tests. The research carried out by this group was direct motivation to expand this project into the tactile modality.

2.4.1 Commercial Introspection

The Peterson tuner BodyBeat Sync ($140) seeks to revolutionize the traditional metronome through its extensive coverage of all three modalities with a wearable pulsing vibration unit which claims to allow musicians to easily internalize the beat and develop a note value relationship both audibly and physically [18].

The Soundbrenner ($99) is a vibration based metronome using an instantaneous pulse and claims that in freeing the ears, it has brought the rhythm closer to the body, making it more comfortable and natural to feel the beat and swing of the music instead of chasing the click [19].

2.5 Summary

These specific haptic ventures have served as foundational works to inform design perspective. The work accomplished by the haptic drumset research group directly served to inform the threshold for acceptable timing of the vibrotactiles. For the four motors in use with the haptic device, a 50 ms ramp up time with 50 ms separation yields a 400 ms (150 bpm) window which agrees well with prior SMS research of rate limits. The vibrotactile metronome gave insight toward test methodology and potential challenges involving latency or real-time strategy. The continuous visual metronome was a solid baseline for data analysis metrics and one of the key inspirational works for the hypothesis of this work.

Chapter 3

Method

This chapter outlines each test case and describes the motivation behind the test plan.

The design and prototype iterations of the haptic metronome are discussed extensively throughout Appendix A.2 along with a parts list and schematic of the hardware builds.[1] The tap hardware construction is described in B.2. A pseudo code breakdown of the test suite design has been offloaded to B.1 along with the round-trip latency system calculation in B.3 which serves to establish a level of confidence in the accuracy of time dependent data.

The overall test principle was derived from traditional sensorimotor synchronization tasks in which a user is asked to tap to a corresponding stimulus. The asynchrony was tracked and plotted along with the *PCR* and any missed taps. Since the haptic domain is of primary focus, the auditory modality functions primarily as a benchmark or baseline foundation. The work presented in 2.4 covers the idea of the interstitial beat occupying the visual domain and as such will not be re-evaluated here.

3.1 Test Plan

Testing was divided into two major sections, **Steady** and **Dynamic**, implying either an *isochronous* or a *non-isochronous* pulse respectively. While structurally identical,

[1]All of the code is open source and readily available at `https://github.com/afaintillusion/he-sm`

Steady										
	Discrete		BPM	Runtime (sec)		Interstitial		BPM	Runtime (sec)
A1a	click	i.	45	20	A1b	legato chime (swing click)	i.	45	30
		ii.	90	20			ii.	90	16
		iii.	135	20			iii.	135	11
		iv.	180	20			iv.	180	8
A2a	staccato music (melody)	i.	45	32	A2b	legato music (melody)	i.	45	32
		ii.	90	16			ii.	90	16
		iii.	135	11			iii.	135	11
		iv.	180	8			iv.	180	8
H1a	poke / all on (instantaneous)	i.	45	15	H1b	oscillate down and back up	i.	45	15
		ii.	90	15			ii.	90	15
		iii.	135	15			iii.	135	15
		iv.	180	15			iv.	180	15
Dynamic									
	Discrete		BPM	Runtime (sec)		Interstitial		BPM	Runtime (sec)
A3a	click	i.	45 +/- 15	20	A3b	legato chime (swing click)	i.	45 +/- 15	20
		ii.	90 +/- 15	10			ii.	90 +/- 15	10
		iii.	135 +/- 15	10			iii.	135 +/- 15	10
		iv.	180 +/- 15	10			iv.	180 +/- 15	10
A4a	staccato music (melody)	i.	45 +/- 15	30	A4b	legato music (melody)	i.	45 +/- 15	30
		ii.	90 +/- 15	15			ii.	90 +/- 15	15
		iii.	135 +/- 15	10			iii.	135 +/- 15	10
		iv.	180 +/- 15	10			iv.	180 +/- 15	10
H2a	poke / all on (instantaneous)	i.	45 +/- 10	15	H2b	oscillate down and back up	i.	45 +/- 10	15
		ii.	90 +/- 5	15			ii.	90 +/- 5	15
		iii.	135 +/- 3	15			iii.	135 +/- 3	15
		iv.	180 +/- 1	15			iv.	180 +/- 1	15

Table 3.1: Test Plan

the dynamic tests however focussed on rubato within a range starting at the predefined BPM and rising or falling within a specified window (maximum span of +/- 15 bpm). The chosen tempi parallels slow walking to running gaits spanning a range of 45-180 beats per minute.

Each section has three subsections centered around an audible metronome tone (**A1, A3**), musical note (**A2, A4**), and lastly the haptic modality (**H1, H2**). Subsections were further broken down into **a** and **b**, denoting either *discrete* or *interstitial*[2] mode of operation. A breakdown of the test plan is shown in Figure 3.1. The data analysis in Chapter 4 will frequently reference this table as a legend.

As discussed in Appendix A.2, the haptic was designed with two operating modes in mind, discrete and continuous. These modes were programmatically controlled to match the desired test cases, extensively explained in section B.1.

[2] Continuous

Group	Instrument	User ID
Amateur	Bass	729
	DJ	390
	Piano	399
	Voice	379
Neither	None	486
		514
		932
	Piano	394
Professional	Flute	410
		591
		824
	Percussion	367
		506
		521
		552
	Piano	510

Table 3.2: Subject Grouping

3.1.1 Subjects

Out of 18 subjects tested, 16 were parsed to equivalently divide the groups into 8 professionals and 8 amateurs/non-musicians. Usernames were anonymized into User ID's using a cumulative char to int conversion method. A breakdown of the grouping per instrumentation is shown in Table 3.2.

3.1.2 Audio File Generation

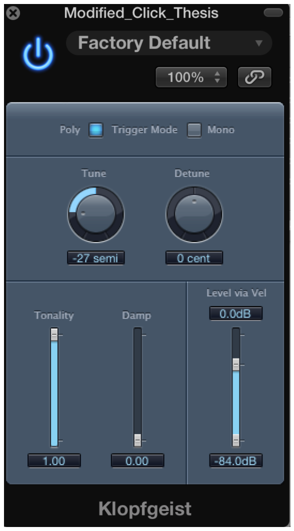
All tracks were rendered using the digital audio workstation (DAW) *Logic Pro X* as *.wav* files at a sample rate of 44.1kHz with 16 bit resolution.

Metronomic click and legato chime

A1a and **A3a** required a standard metronomic pulse. This was accomplished using the default Klopfgeist (metronome) plugin from Logic Pro X. No additional tuning was modified and the tonality was left at 0.83 of unity.

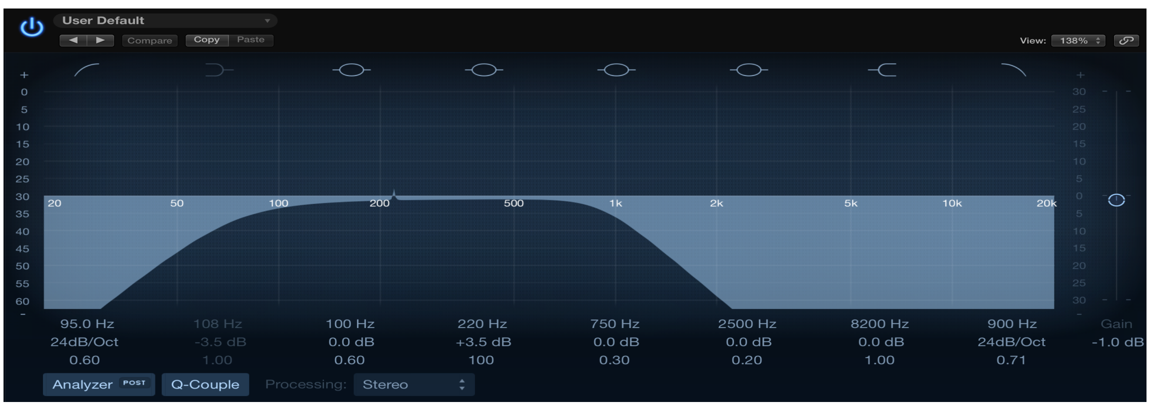
Figure 3.1: Modified click parameters for interstitial tests.

(a) Modified metronome



(b) Superimposed tremolo

(c) Equalized tone

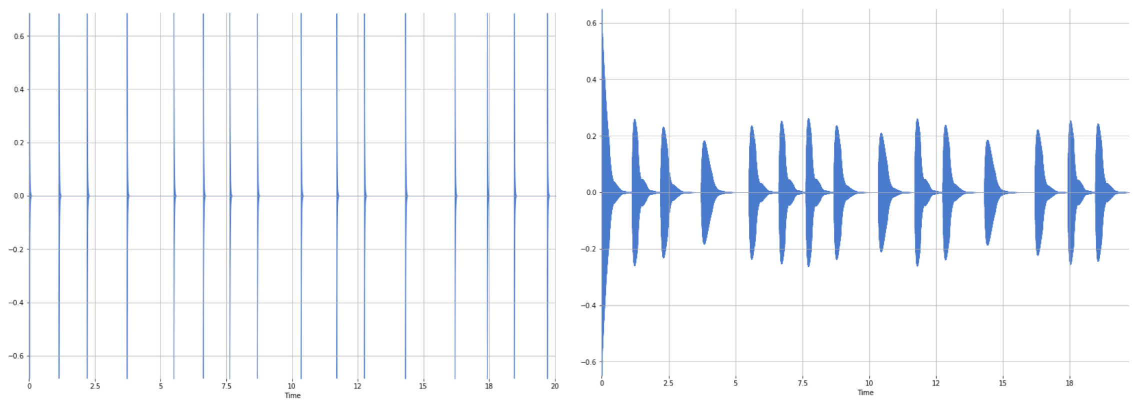


A1b and **A3b** however required a swing or legato type of chime in order to convey filling the interstitial space. To capture this effect the Klopfgeist tonality was increased to unity and tuned -27 semitones lower which served to both soften diminish the discrete click and provide an elongated or continuous audible sensation. To give the impression of a sound that was ramping up in amplitude and decaying after the peak, a sawtooth wave was added to the signal chain as shown in Figure 3.1. Last, a multi-band EQ was placed at the end of the signal chain with a bandpass filter from 95Hz-750Hz removing any unwanted frequency presence with a 3.5dB high-Q peak at 220Hz to emphasize the tonality.

The resultant waveform encapsulated the occupation of the interstitial space. A comparison of this waveform in contrast to it's discrete counterpart is shown in 3.2. Note the envelope of signal (b) follows a natural build up and decay.

Figure 3.2: Metronomic waveform comparison

(a) A3a1: discrete audible click (b) A3b1: interstitial tone

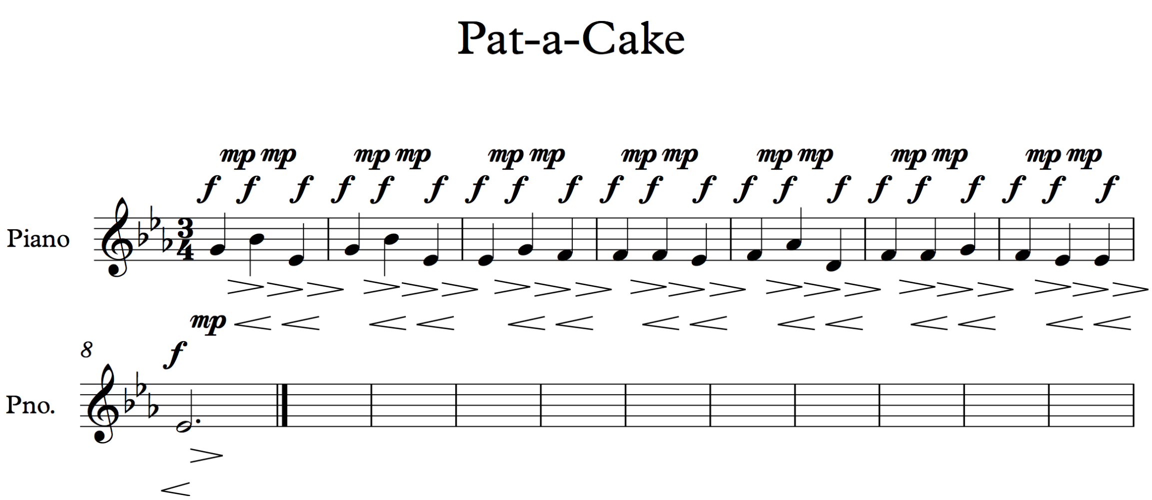


Stacatto and legato melody

As a specific musical listening task, test cases **A2a**, **A4a** and **A2b**, **A4b** involve synchronization to a simple melodic sequence of notes. The music chosen was the nursery rhyme *Pat-A-Cake*. The initial mockup was drafted in Sibelius and exported to Logic Pro X for bpm adjustment.

Each quarter note represents a beat and therefore a 1:1 synchronization tap onset task. In order to emphasize a discrete event for test cases **A2a** and **A4a**, notes were input as stacatto, shown below:

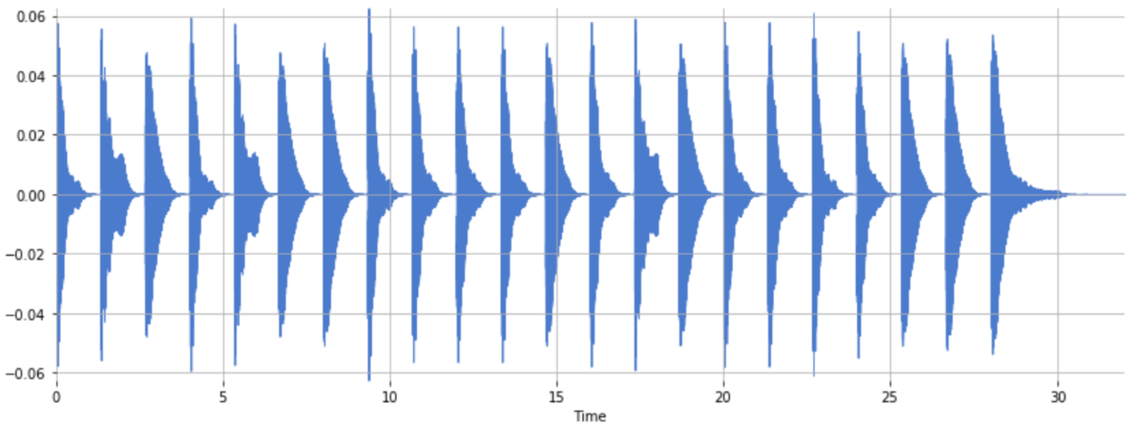
The interstitial counterparts (**A2b**, **A4b**) similarly underwent crescendo and decrescendo after every note onset with forte accents surrounded by mezzopiano to give the impression of amplitude build up and decay[3], shown below:



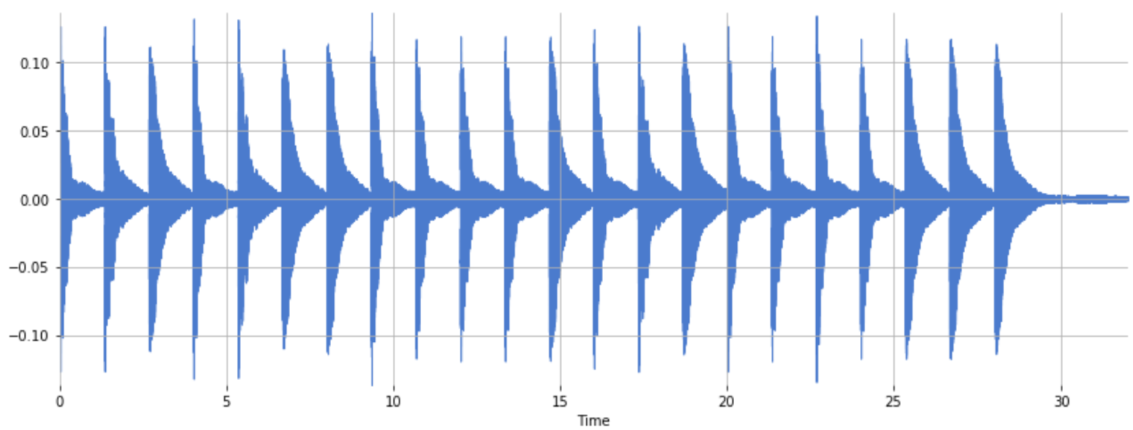
[3]Note in Figure 3.3 the gradual, nearly exponential decay displayed in the interstitial tone as a result of the legato input along with the amplitude difference due to the forte accents.

Figure 3.3: Musical waveform comparison

(a) A2a1: stacatto melody



(b) A2b1: legato melody

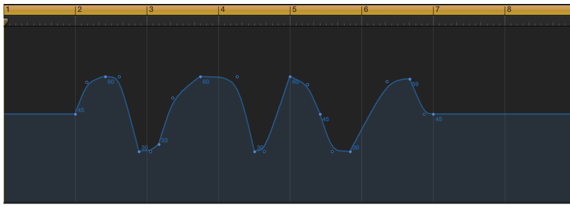


Dynamic tempi manipulation - audio

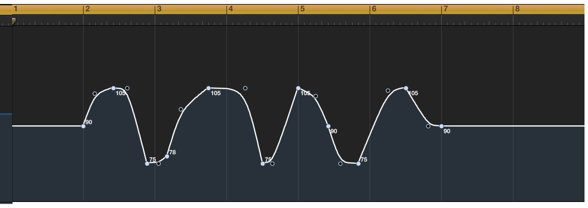
Dynamic manipulation of tempo was accomplished in *Logic Pro X* through automation of the tempo parameter over the time period of the desired waveform. Each test case started on one of the pre-defined BPM's (45, 90, 135, 180) but traversed either sinusoidally or triangularly through segmented time blocks as peaks and troughs ranging plus or minus 15 bpm; shown in 3.4.

Figure 3.4: Dynamic audio tempo automation patterns

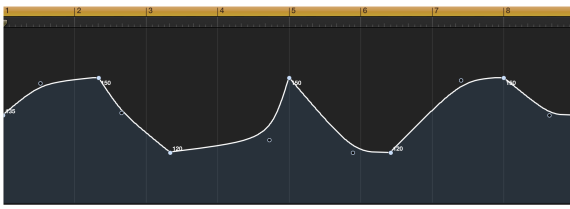
(a) 45 +/- 15



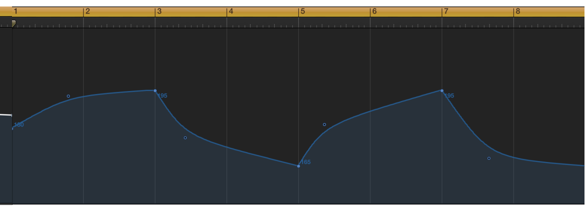
(b) 90 +/- 15



(c) 135 +/- 15



(d) 180 +/- 15



3.2 Test Suite

High precision data acquisition and the minimization of delay were the central foci of the test suite design. Due to the extensive amount of publicly available libraries, multithreading capability, Pandas[4] dataframe structure, and plot integration via matplotlib, *Python*[5] was chosen as the development environment. Complementary to the software platform was the implementation of a tap onset detection mechanism via force sensitive resistor (FSR) on the *Arduino Uno*[6].

For further detail, including a pseudo-code breakdown, please see B.1.

[4] https://pandas.pydata.org/
[5] https://www.python.org/
[6] https://store.arduino.cc/usa/arduino-uno-rev3

Chapter 4

Data Analysis

This chapter analyzes the data from the 16 users tested and draws conclusions based on the proposed hypothesis. It begins with a brief overview of the test conditions followed by the main results analysis in 4.2. This includes validation of findings from past sensorimotor synchronization research and the claims mentioned in 1.3.3.

4.1 Overview

As previously mentioned, the group was split into 8 professionals[1], 8 non-musicians and amateurs. The level of musical experience varied, as seen in Figure 4.1, from less than 1 year to over 10 years. Each user was self-classified as either a Professional, Amateur, or Neither via questionnaire at the end of the test.

[1]A special thank you is in order to the professional musicians of the Army Old Guard Fife and Drum Core for volunteering their time.

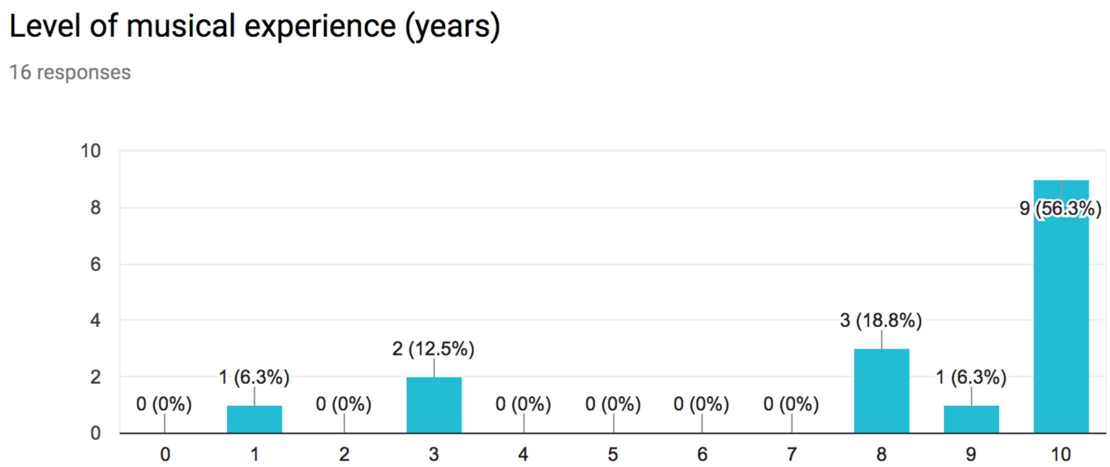


Figure 4.1: Musical Experience

The instrumental breakdown can be seen in Figure 4.2 below.

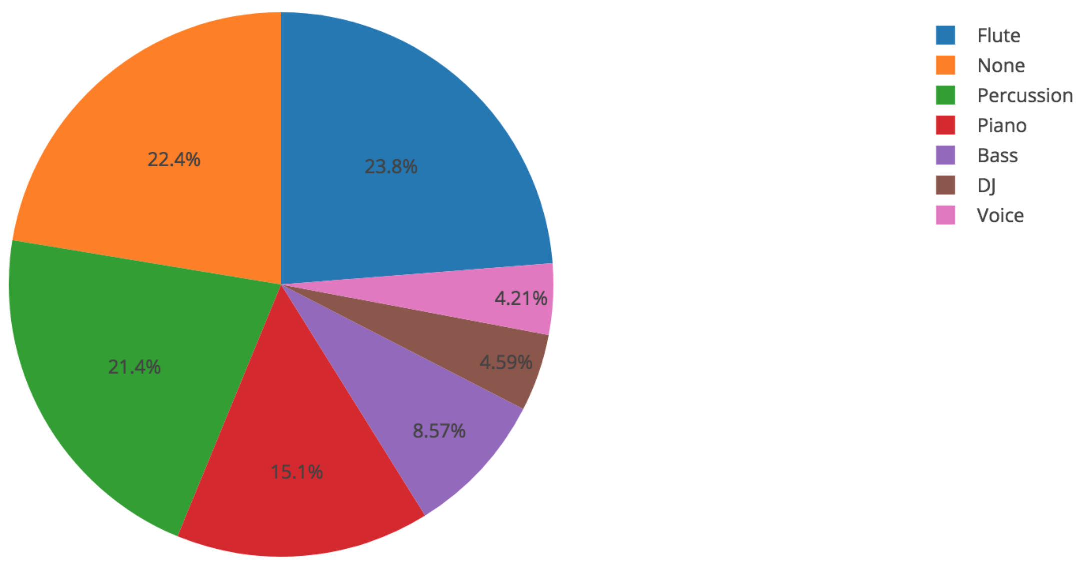


Figure 4.2: Spread of instrumental ability (out of 16)

Thirteen users had some sort of prior experience with an audible metronome and three did not. For most this was their first experience with a haptic device. All users chose their dominant arm to wear the haptic sleeve and chose their dominant hand for tapping.

4.1.1 Setup

To initialize setup, the subject is seated and given a pair of closed-back headphones. The FSR is situated to their preference and secured into place. Unlike a keyboard or button the FSR gives little to no feedback or rebound. This ensures a confident tap on each onset while providing no tactile response. The approach seeks to avoid intrinsic lag due to its independence of mechanical components. The delay limit is defined by the threshold applied in the software to avoid debounce, as discussed in Section B.1

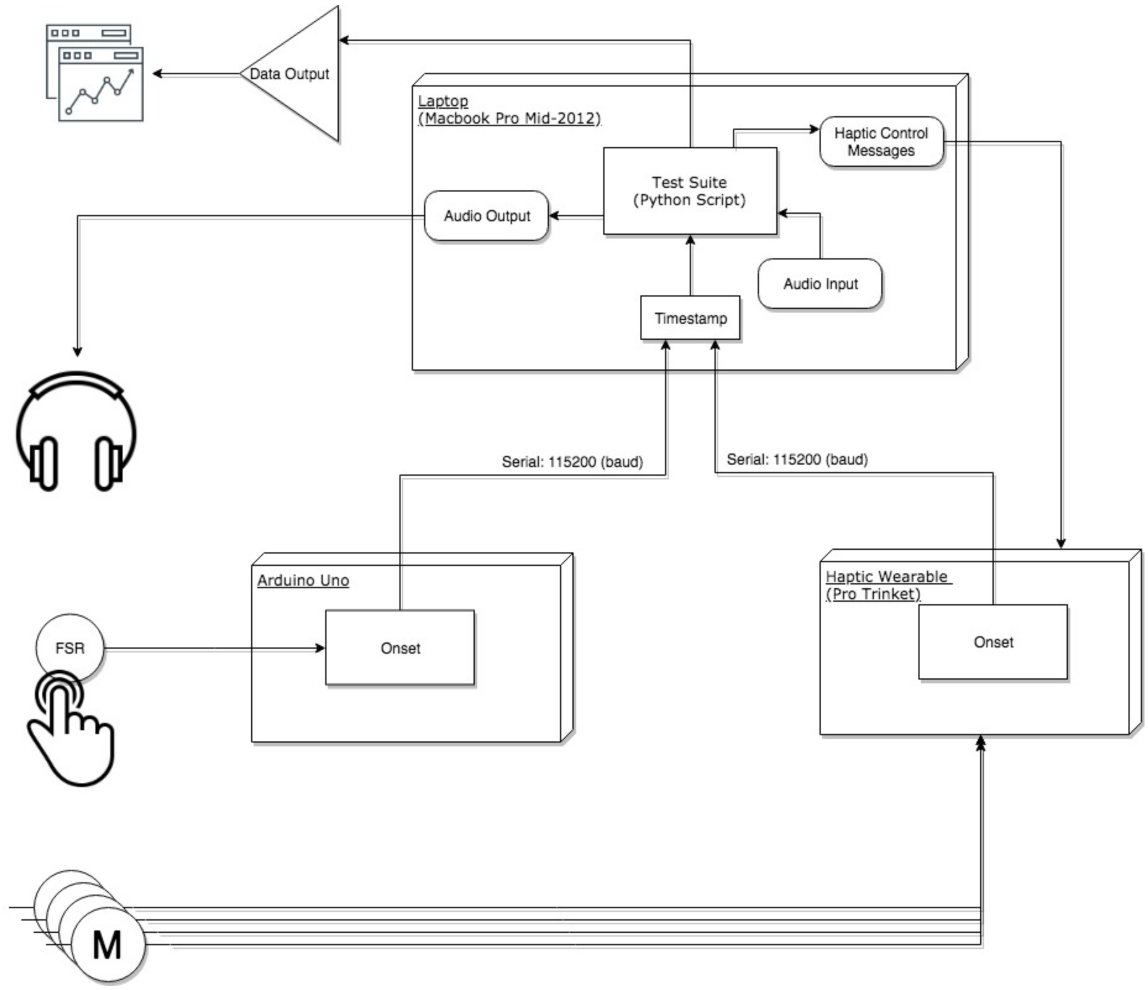


Figure 4.3: Test Suite Flow Chart

The Python file *testSuite.py* is run and the UI will prompt the subject to input their name, read the instructions, agree to the conditions of the test suite, and commence with the test. The first 8 are practice tests to get used to the haptic sensation as well as the variety of audible test cases. The order of test case execution is scrambled with a static seed pseudo-random generator such that every user encounters the same test order. Every iteration of the test plots the Tap Onset, True Onset, and Sanitized Onset for the purposes of feedback and affirmation of correct tapping as seen in Figure 4.4. Upon completion of the 48 test cases, the users are asked to fill out a survey for feedback.

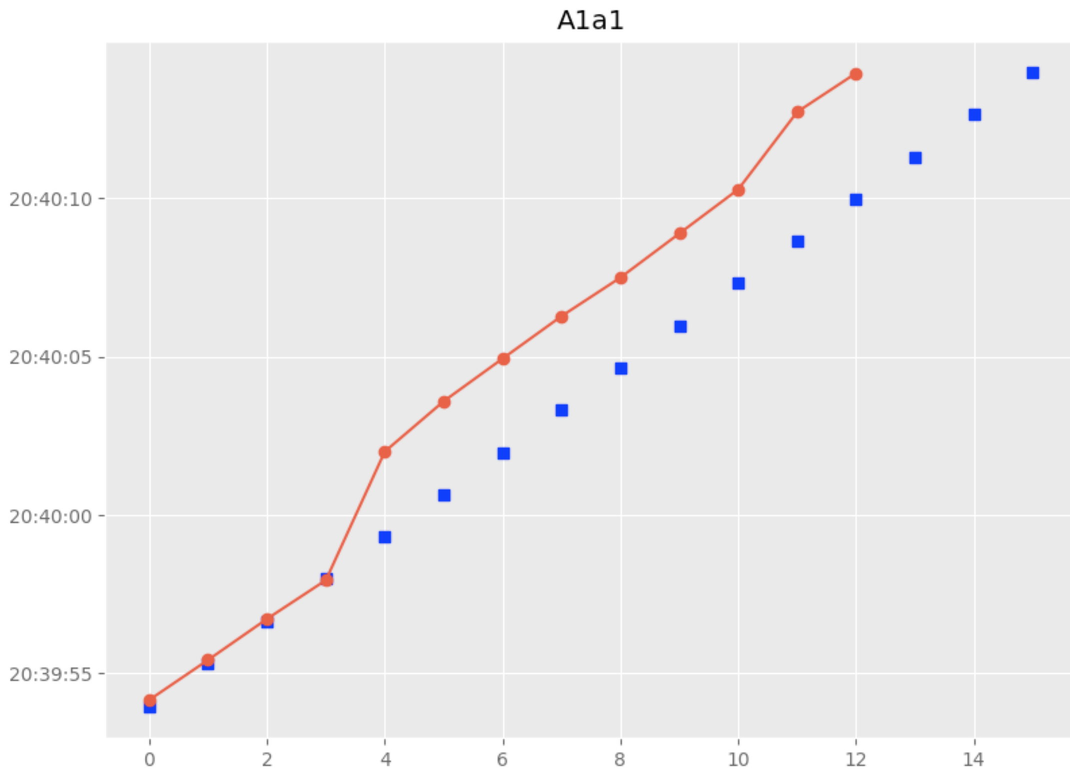


Figure 4.4: Test Case Feedback Example: Red circles are the tap onset, blue squares are the true onset. The x-axis is the tap count, the y-axis is elapsed time.

4.2 Results

The main metric of stability and overall performance, as discussed in Section 1.3.1, is the asynchrony. In all test cases a sanitization procedure was constructed to ensure that a missed tap would allow for the user to get back on track with the true onset later in the dataset - a process detailed entirely in Section B.1.8. Finally, a latency correction factor was applied dependent on whether the test case was audible or haptic based. For more information on the derivation of this constant see Section B.3.

4.2.1 SMS Research Expectations

Prior SMS research has set expectations for the audible modality such that as the IOI increases the standard deviation of asynchrony (SD_{asy}) is expected to increase non linearly. This means that as we approach a slower bpm, we would anticipate larger deviations from the mean. This claim in in agreeance with the dataset across both modalities as shown in Figure 4.5 when viewing a subset of four test cases from right to left. For example, A3a1 relative to A3a2 through A3a4 and H1a1 compared to H1a2 through H1a4. These were test cases which had the highest inter-onset-interval and clearly exhibit high levels of variation relative to their adjacent test cases.

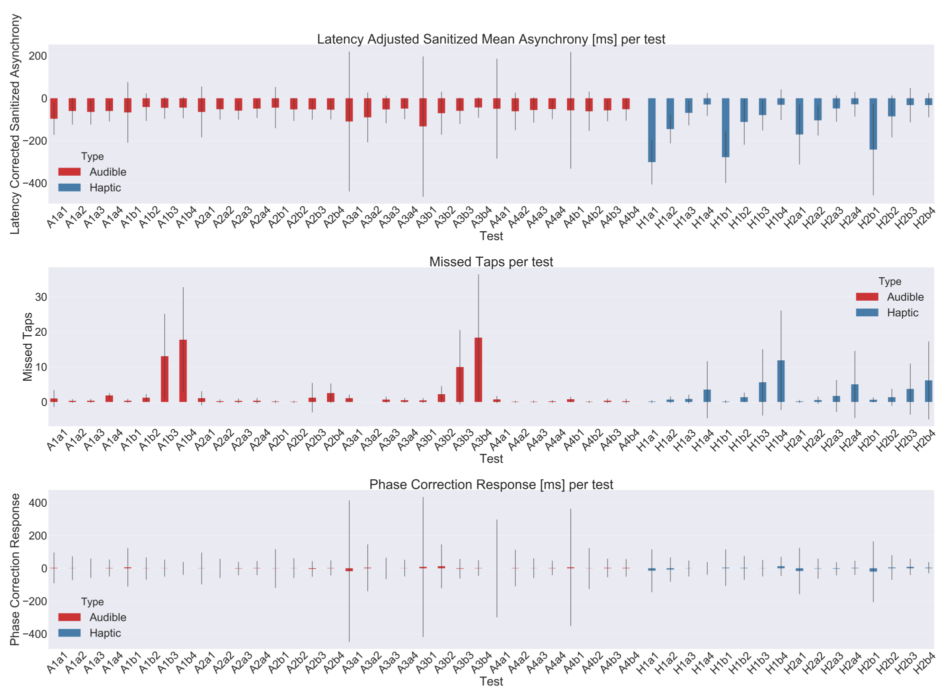


Figure 4.5: Summary across all participants

The black lines represent the standard deviation. The dynamic audible tests, A3a1 through A4b4, exhibited great deviation in both *Latency Corrected Sanitized Asynchrony* (LCSA) and *Phase Correction Response* (PCR), or the variation of one tap onset from the next. The amount of metronomic jitter was varied on each repetition for these test cases and thus a reactionary response is expected since there was no real method of taking a proactive approach for these particular tests.

Most taps were missed during the interstitial legato chime or swing click A1b3, A1b4, A3b3, A3b4 as well as during the ramp up haptic test cases H1b3 and H1b4. These test cases ranged from 120 bpm to 180 implying either an indistinct onset, overstimulation, or low signal to noise ratio.

Group Trends

SMS findings anticipate a lower (SD_{asy}) for professionals compared to non musicians and expect little to no difference between amateurs and non musicians. As seen in Figure 4.6, the results agree with prior research. Professionals had a mean LCSA of approximately -45 ms across all audible tests and approximately -75 ms for haptic tests relative to -55 and -90 ms for amateurs and -90 and -80 ms for non-musicians.

The non-professional group did marginally better on the haptic test cases than audible, though with a higher amount of missed taps. The standard deviation of the phase correction response, shown at the bottom of the figure, implies an ability to adapt tapping synchronization from beat to beat more successfully with the haptic device than across audible tests. This lends very promising insight but must be further classified to determine if the benefit is test case dependent or involving of strictly non-isochronous beats.

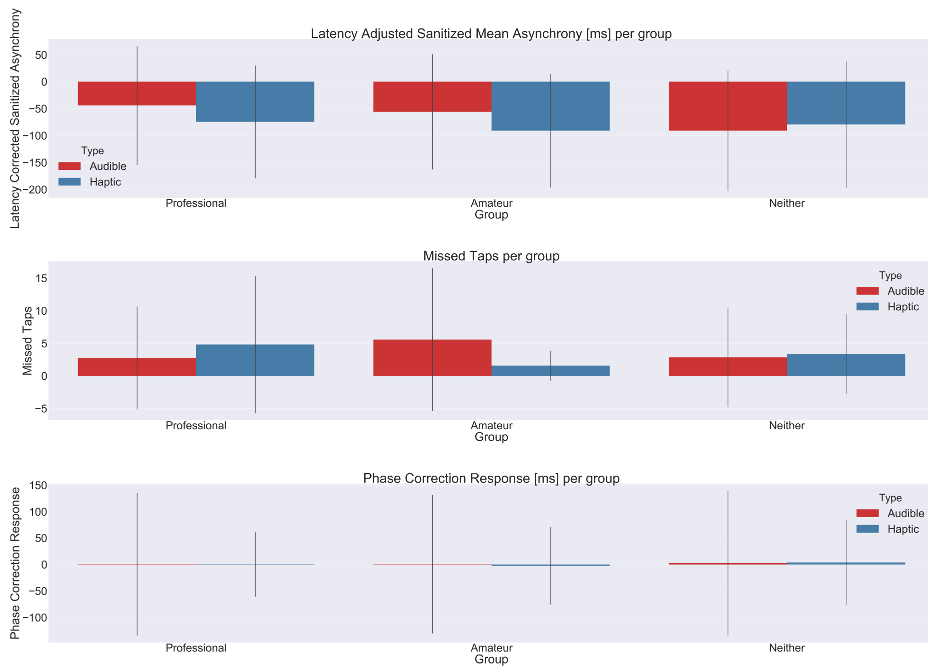


Figure 4.6: Group test summaries

4.2.2 Instrumental Breakdown

Past research into sensorimotor synchronization suggests a heightened rhythmic sensibility within percussionists and pianists who notoriously have held lowest variability values for tap tests. Figure 4.7 would suggest agreement with the exception of Bass. This is however, inconclusive, since the bins are uneven. For instance, whereas there was only a single bassist in the test, there were four percussionists. Without accounting for missed taps, it would appear that flutists held the lowest asynchrony across both modalities but the flutists were the group with the most missed taps for the haptic device. This could imply a negative interaction involving placement as these musicians are accustomed to a level of freedom on their arms.

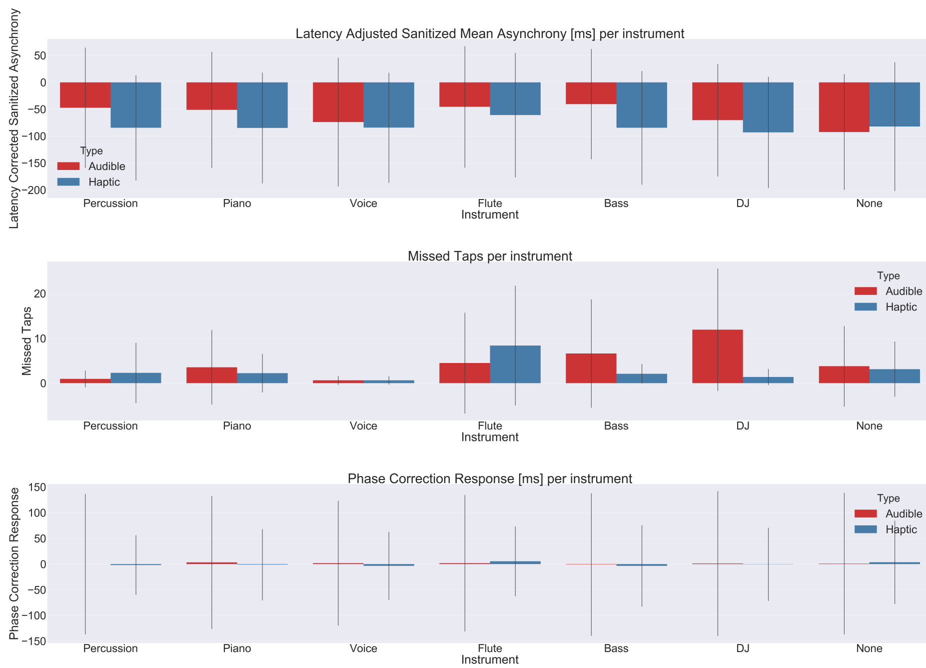


Figure 4.7: Summaries per instrument

4.3 Haptic versus Audible Tests

As expected, audible tests are the clear winner across all participants for the static test cases. As shown in 4.8, the mean of the audible tests is −56.92 ms with a standard deviation of 69.51 as compared to a mean of −92.67 ms with a much larger standard deviation of 109.29 for the haptic tests. With a delta of 35.75 ms between this would suggest that users are slightly more comfortable responding to isochronous beats with the auditory modality.

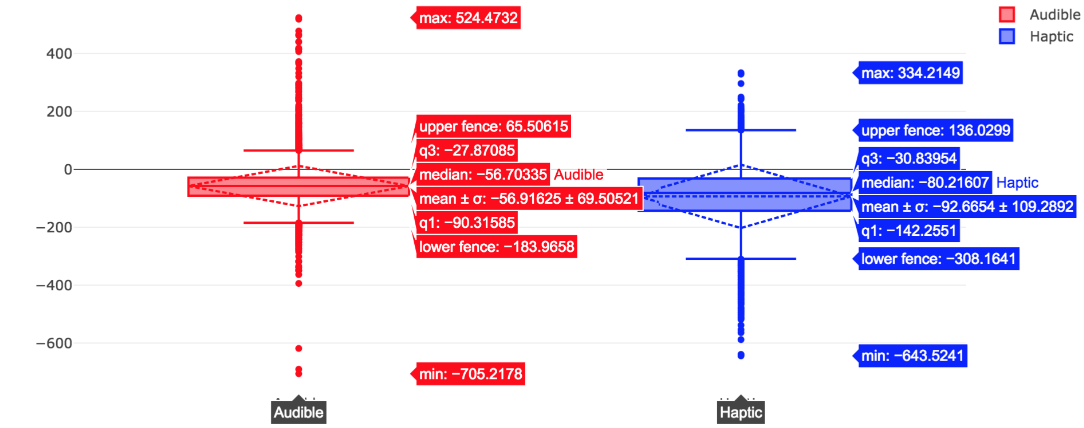


Figure 4.8: Latency Corrected Sanitized Asynchrony for Static Tests

The results are much more promising when comparing the dynamic test cases. As seen in Figure 4.9, the haptic test results were a negligible 7.38 ms away from the audible but with a much cleaner standard deviation. This implies a greater synchronization ability with the haptic device and strongly supports the hypothesis of this work.

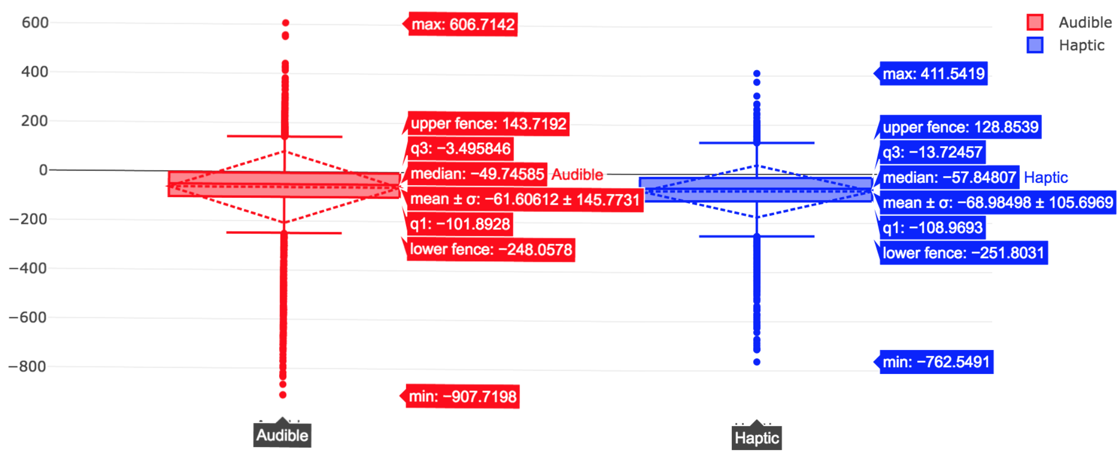


Figure 4.9: Latency Corrected Sanitized Asynchrony for Dynamic Tests

In a breakdown of asynchrony for static versus dynamic tests per group as shown in Figure 4.10, the dynamic haptic test cases exhibit a shift towards zero implying an improvement in synchronization during non-isochronous beats.

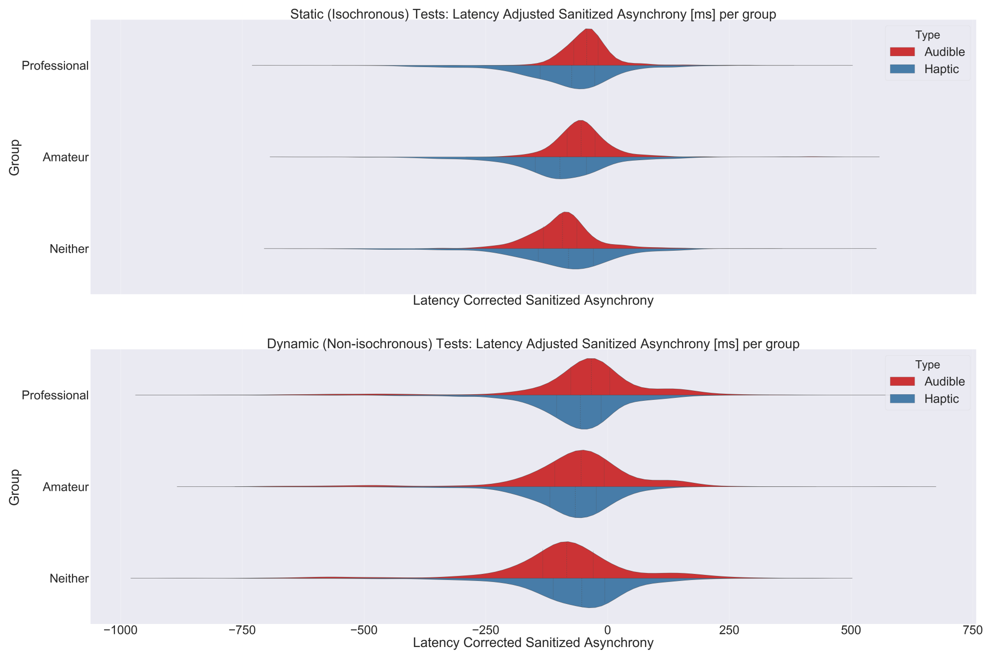


Figure 4.10: Kernel Density Estimation: Latency Corrected Sanitized Asynchrony for Static vs Dynamic Tests

This finding heavily supports the overarching hypothesis of this work and can be further exemplified in Figure 4.11 and 4.12 below. As we traverse from left to right on the graph, the IOI increases - representing a decreasing bpm. The trajectory is nearly flat for the static audible test cases in red at the top.

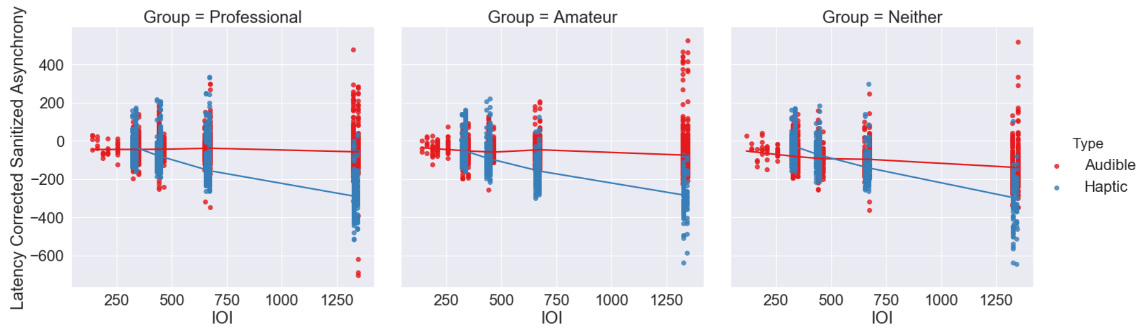


Figure 4.11: Latency Corrected Sanitized Asynchrony vs. Inter-Onset-Interval across static test cases.

However, the dynamic audible tests trend towards a positive asynchrony implying a reactive approach. Near slower tempi the haptic tests contrarily trend steeply toward negative asynchronies in both the static and dynamic test cases. The deviation implies a proactive approach. The implications hint at an overall success for users to concretely navigate non-isochronous beats with the haptic metronome device.

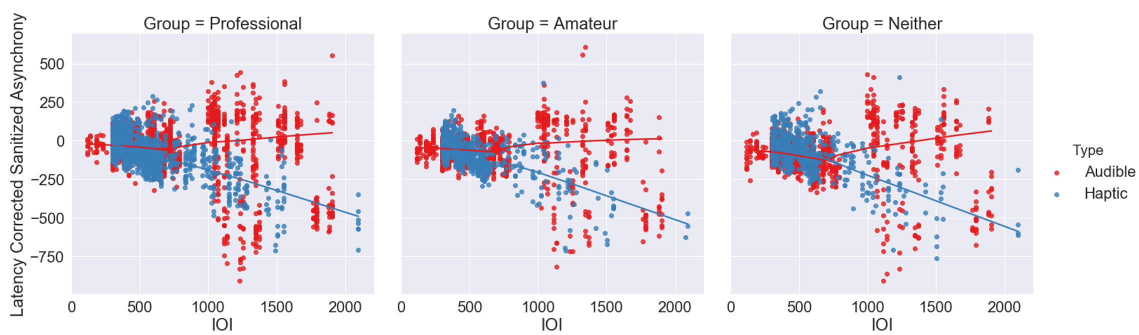


Figure 4.12: Latency Corrected Sanitized Asynchrony vs. Inter-Onset-Interval across dynamic test cases.

4.3.1 Haptic Poke versus Haptic Ramp

To compare whether users were more inclined towards a particular mode of operation, either all on all off or ramp up ramp down, of the haptic and to check whether filling the interstitial space with the ramping sensation had any impact, the results in Figure 4.13 contrast the static and dynamic haptic instantaneous or poke test results to the haptic ramp method. The static haptic poke had an asynchrony of -93.17 +/$-$

105.16. The static haptic ramp had an asynchrony of −92.12 +/− 113.53. The delta is negligible and no conclusion can be made for a benefit within the isochronous realm for the haptic ramp. The dynamic haptic poke had an asynchrony of −69.93 +/− 90.18. The dynamic haptic ramp had an asynchrony of −67.94 +/− 120.43. A slightly better delta for the ramp sensation but a larger deviation implies that most users preferred the instantaneous sensation.

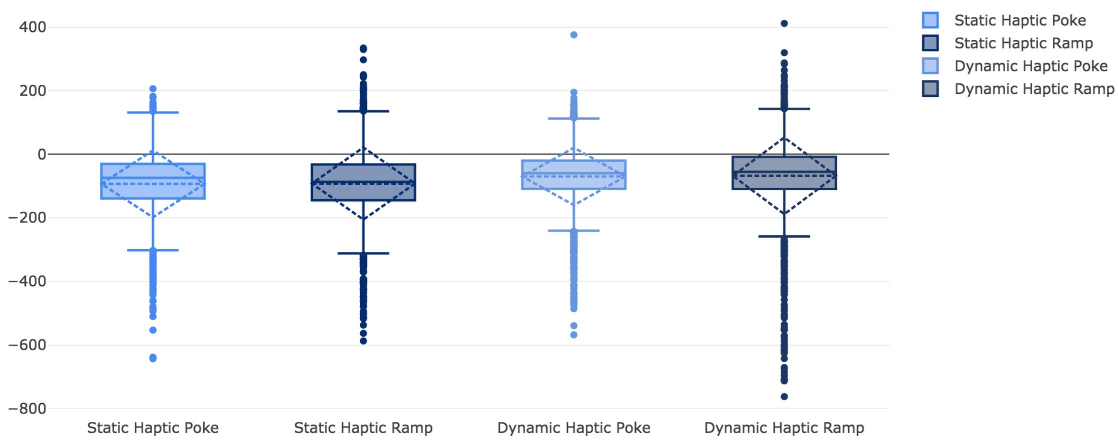


Figure 4.13: Latency Corrected Sanitized Asynchrony of haptic poke versus ramp effect across both static and dynamic tests.

4.3.2 Audible Click versus Audible Music Tone

To find out whether the legato audible music tone which saught to fill the interstitial space had any sort of favorable impact a mean comparison was conducted as shown in Figure 4.14. The static audible click had a mean asynchrony of −59.47 with a deviation of 71.08 while the static audible musical tone was −53.46 +/− 67.16. These results are so close that no improvement can be claimed.

The dynamic audible click had a mean asynchrony of −67.84 +/− 152.39 while the dynamic audible musical tone had a mean asynchrony of −55.73 +/− 138.98. From these results it would appear that there was a slight advantage when participants heard a musical tone that filled the interstitial space as opposed to a discrete click.

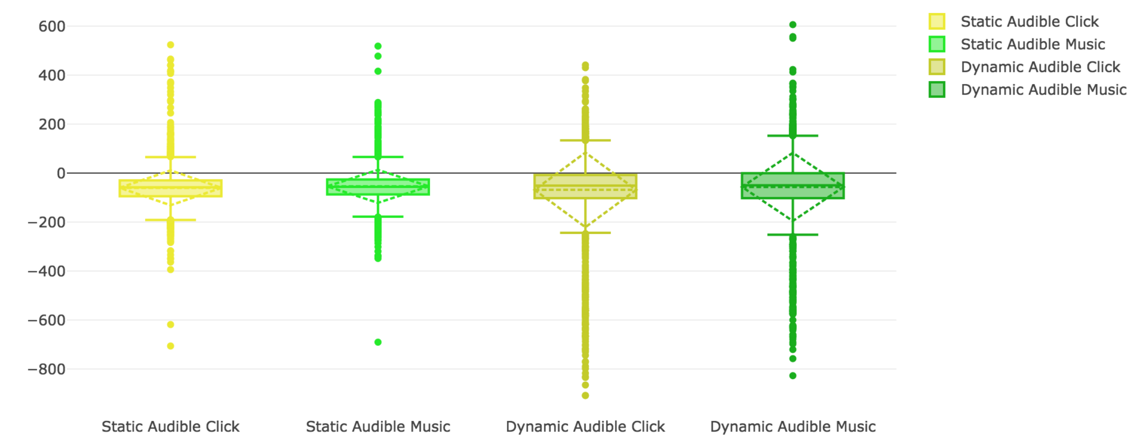


Figure 4.14: Latency Corrected Sanitized Asynchrony of audible click versus music tone across both static and dynamic tests.

4.3.3 Analysis of variance across test cases

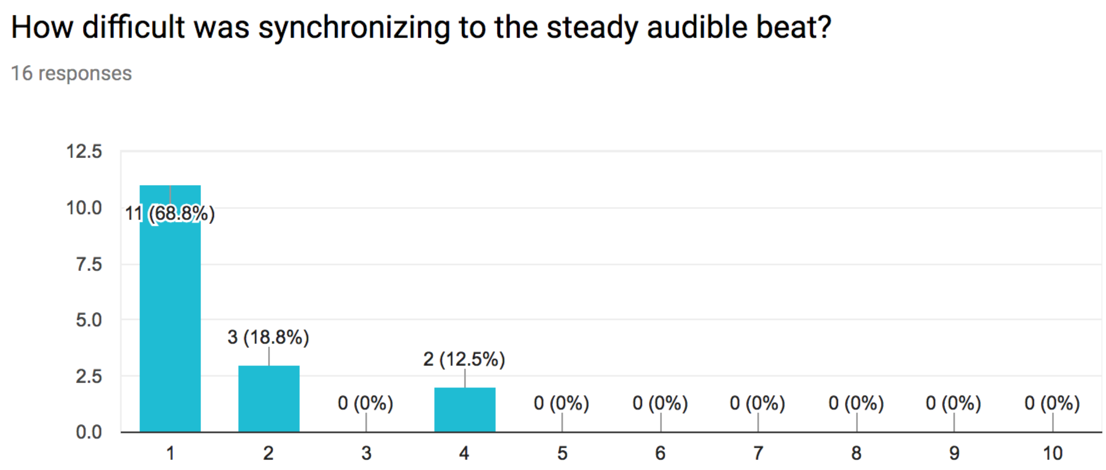
To confirm the significance of the differences found between haptic and audible tests, an independent t-test with unequal variance was conducted. The null hypothesis was the assumption of no difference between these test case types in the latency corrected sanitized asynchrony value. P-values were obtained for haptic versus audible test case combinations across the static and dynamic groupings. The results are summarized in Figure 4.15.

```
Static Test Cases: A1 and A2 vs. H1
The results of the independent t-test are:
        t-value = -16.968
        p-value = 0.000

The difference between groups is -35.7 [-39.5 to -32.0] (mean [95% CI])
```

```
Dynamic Test Cases: A3 and A4 vs. H2
The results of the independent t-test are:
        t-value = -2.732
        p-value = 0.006

The difference between groups is -7.4 [-12.8 to -2.0] (mean [95% CI])
```

Figure 4.15: T-Test Results

4.4 Feedback



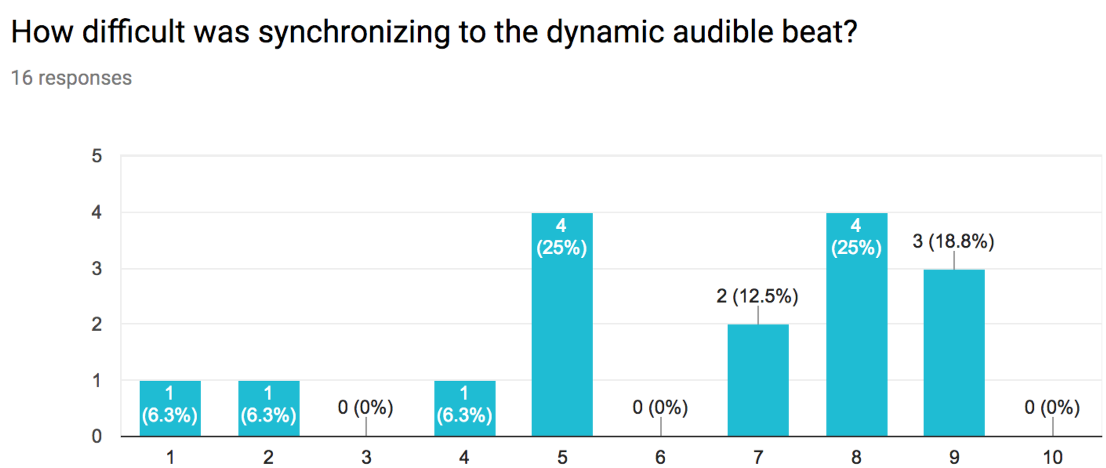
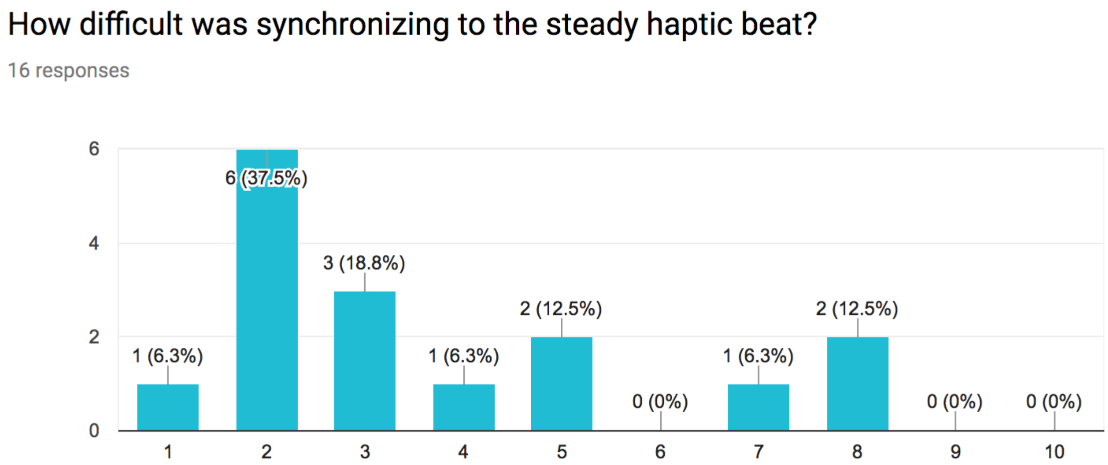


Figure 4.16: Questionnaire: Difficulty results for dynamic audible tests.

On a scale of 1 to 10, with 10 being the most difficult, approximately 70% of those tested found synchronization to the steady audible beat to be a level of 1, or extremely easy. The remaining 30% found it to be either a 2 or 4 level of difficulty. However, the spread for dynamic audio test cases was much wider ranged, with the majority expressing a high level of difficulty 81.3% above level 5 seen in Figure 4.16. The haptic tests had a wider difficulty spread across the steady beat shown at the top of Figure 4.17. This was to be expected as users had no prior experience with this haptic device let alone any other sort of wearable metronome. If retested or trained over the course of a few weeks to the sensation of touch the responses might have been more favorable or closer in resemblance to the static audible tests. Regardless, the dynamic beat for the haptic modality yielded less difficulty rankings than the dynamic audible (11 ranked a level of 5 or more difficulty vs. 13 for audible dynamic tests), which further supports the hypothesis of this work.



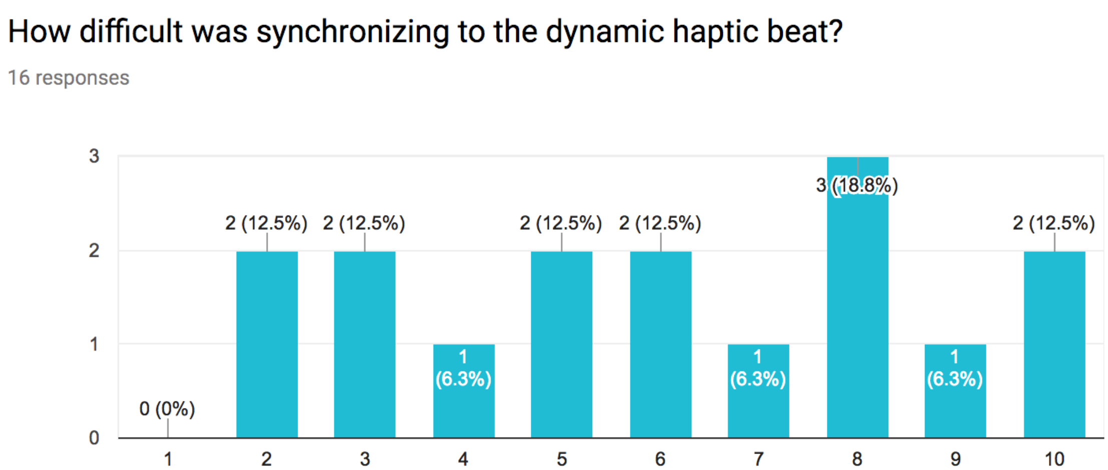


Figure 4.17: Questionnaire: Difficulty results for dynamic haptic tests.

With the question of modality preference, auditory won hands down for the steady beat. Some users would have preferred a combination of both auditory and haptic, though the option was not exemplified in the test suite. The haptic won by 13% for the dynamic tests, shown in Figure 4.18

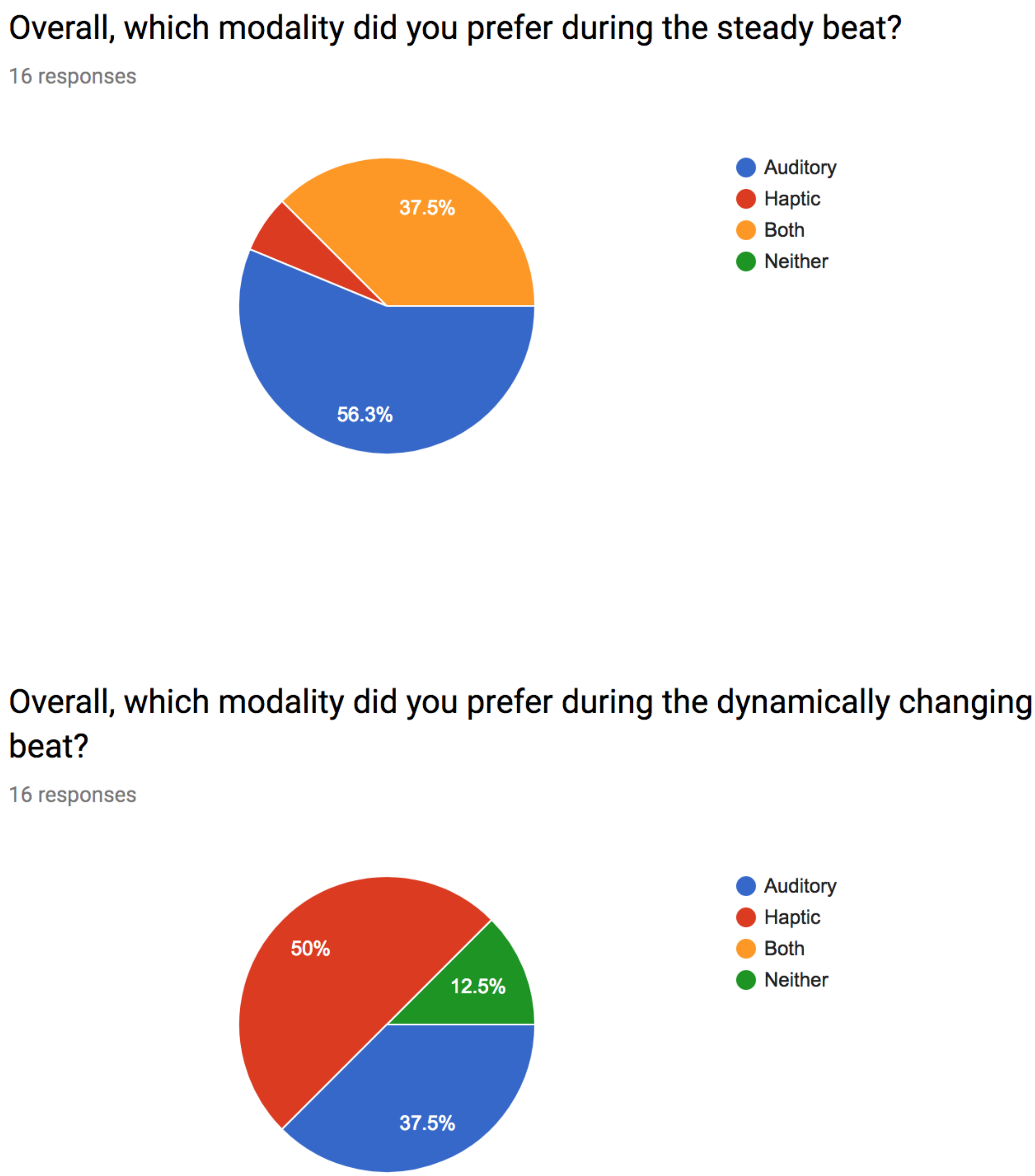


Figure 4.18: Questionnaire: Modality preference

When asked specifically about the preference for haptic mode of operation, it seemed that most users preferred the all on all off mode see in Figure 4.19

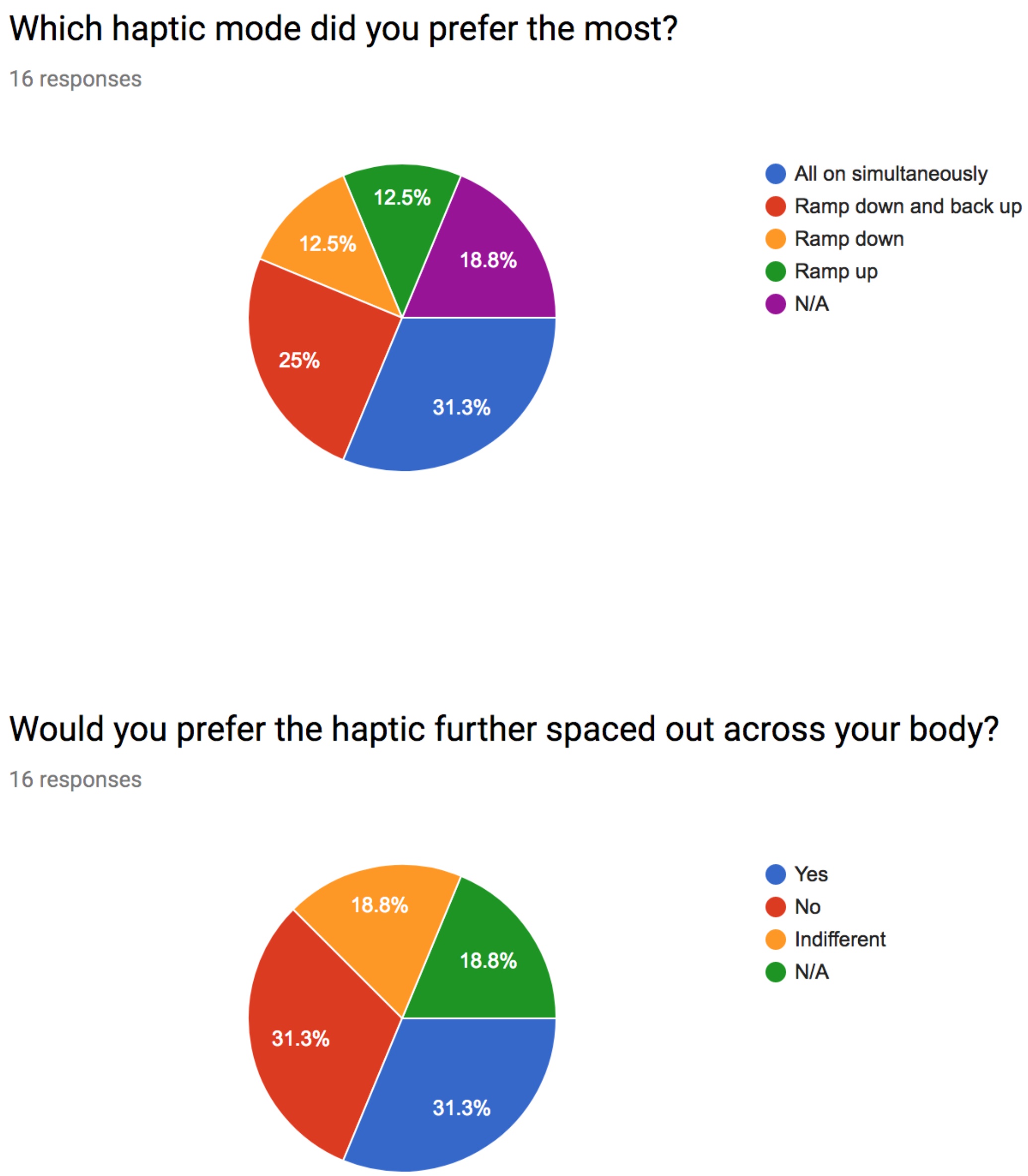


Figure 4.19: Questionnaire: Haptic mode preference and spacing

This could be attributed to the lack of stimulation strength at higher vibrating frequencies for the ramping mode, an intrinsic flaw due to the nature of power operation of the haptic prototype at 5 V 500 mA laptop USB max output. This would be remedied in the next design iteration discussed in 5.1.2.

Furthermore, the users were tested at up to (and occasionally past) 180. This is well beyond the documented 150 bpm limit. At such rapid speeds the vibrotactiles had little time to ramp up to full capacity. This was another reason for diminished strength and could have possibly yielded a sensation nearly indistinguishable from noise.

When asked about placement preference, there was a 31.3% split between the desire to have it further spaced or not. Prior research advocates a larger area of coverage to isolate stimulation zones and promote perceptivity. This would be adopted in a more flexible way in the next prototype iteration with velcro straps spaced out to the desired length.

4.5 Conclusions

Overall, the results are quantitatively promising in support of the hypothesis that filling in the interstitial space provides benefit for non-isochronous or dynamic beats.

Chapter 5

Conclusions

5.1 Future Work

5.1.1 Design Commentary

Much effort went into providing an important framework for the haptic to operate smoothly from a hardware standpoint but the most important progress with the final prototype was the solidification of the software framework. With the integration of delay independent state-machines performing synchronous tasks, haptic mode selection, and precise bpm control, the vibrotactile haptic could enter the next phases of development.

The addition of BLE capability to issue commands over a paired interface such as an app on a smartphone, custom PCB to minimize form factor, and independent node abstraction of each vibrotactile could be realized. This would allow wearers capability to place a programmable array anywhere on their body enabling wide surface area dispersion and source separation.

Lastly, experimentation with other types of vibrotactiles; either the Tactor, TacHammer, or LNA's would be more optimized for touch applications with cleaner signals, dynamic range, and optimized for skin resonance (250Hz).

5.1.2 Wireless Prototype

Over the course of summer of 2018, a wireless prototype was developed which eliminated the FTDI over USB interface and was completely reliant on battery power. A 9 V was connected to a voltage regulator supplying up to 5 V for the motors and 3.3V for the new MCU, sourcing up to 500 mA/hour with enough isolation to have a significantly stronger motor vibration. The MCU was based on the Particle[1] ecosystem, a cloud based IoT framework, and granted BLE and wireless capability.

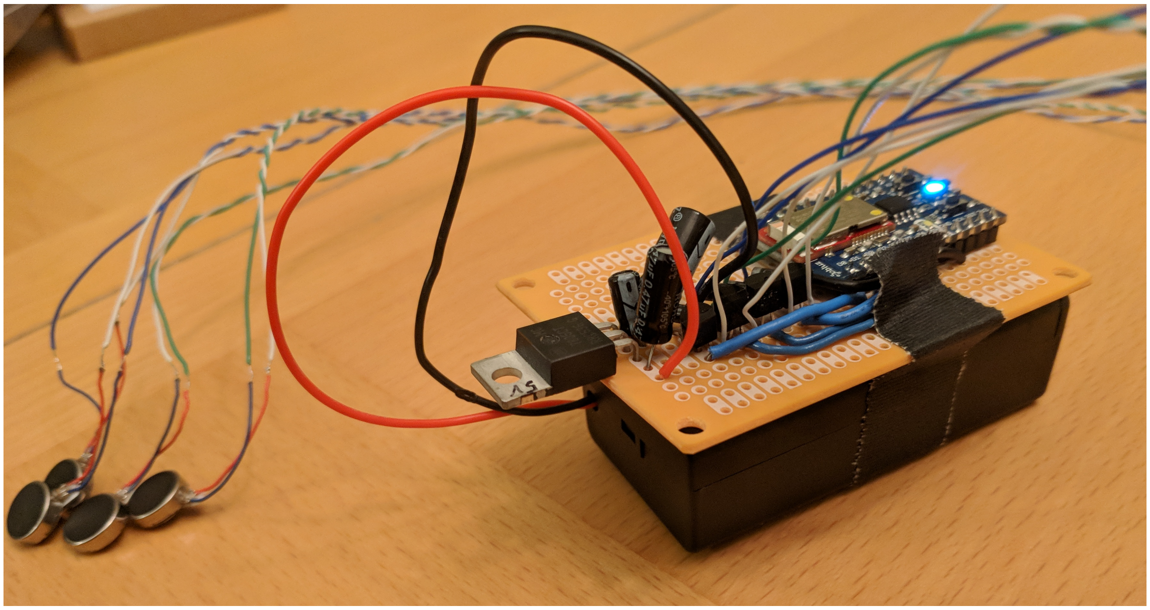


Figure 5.1: Latest wireless prototype with cloud capability

The code was altered to adapt the publish/subscribe methodology where simple numeric BPM values could be passed as API calls to the device given proper OAuth2 authentication. For example, to start the motor at 60 bpm tempo:

```
curl https://api.particle.io/v1/devices/<deviceID>/motor \
  -d arg="1" \
  -d access_token=<accessToken>

curl https://api.particle.io/v1/devices/<deviceID>/motor \
```

[1] https://www.particle.io/

```
-d arg="60" \
-d access_token=<accessToken>
```

5.1.3 Real-time Beat Tracking

This work served as the foundational layer for the real-time beat tracking integration that is to follow. The goal is to have a haptic environmental sensing metronome, one which would intelligently adapt to the dynamics of its surroundings. Some preliminary research has been conducted to get a glimpse of the viability of such a system.

As part of the PhD work completed at the Queen Mary University of London, Adam Stark has developed a causal beat tracking algorithm intended for use in live performances[2][20]. The real-time beat tracking results were comparable to top of the line non-causal systems. The algorithm was initially implemented in C++ with wrappers for Python but externals were later made available for Max/MSP. The Max patch was expanded to fit the context of this work. In doing so the patch was retrofitted to send instantaneous BPM signals over serial thus communicating with the haptic device shown in Figure 5.2.

[2] http://adamstark.co.uk/project/btrack-a-real-time-beat-tracker/

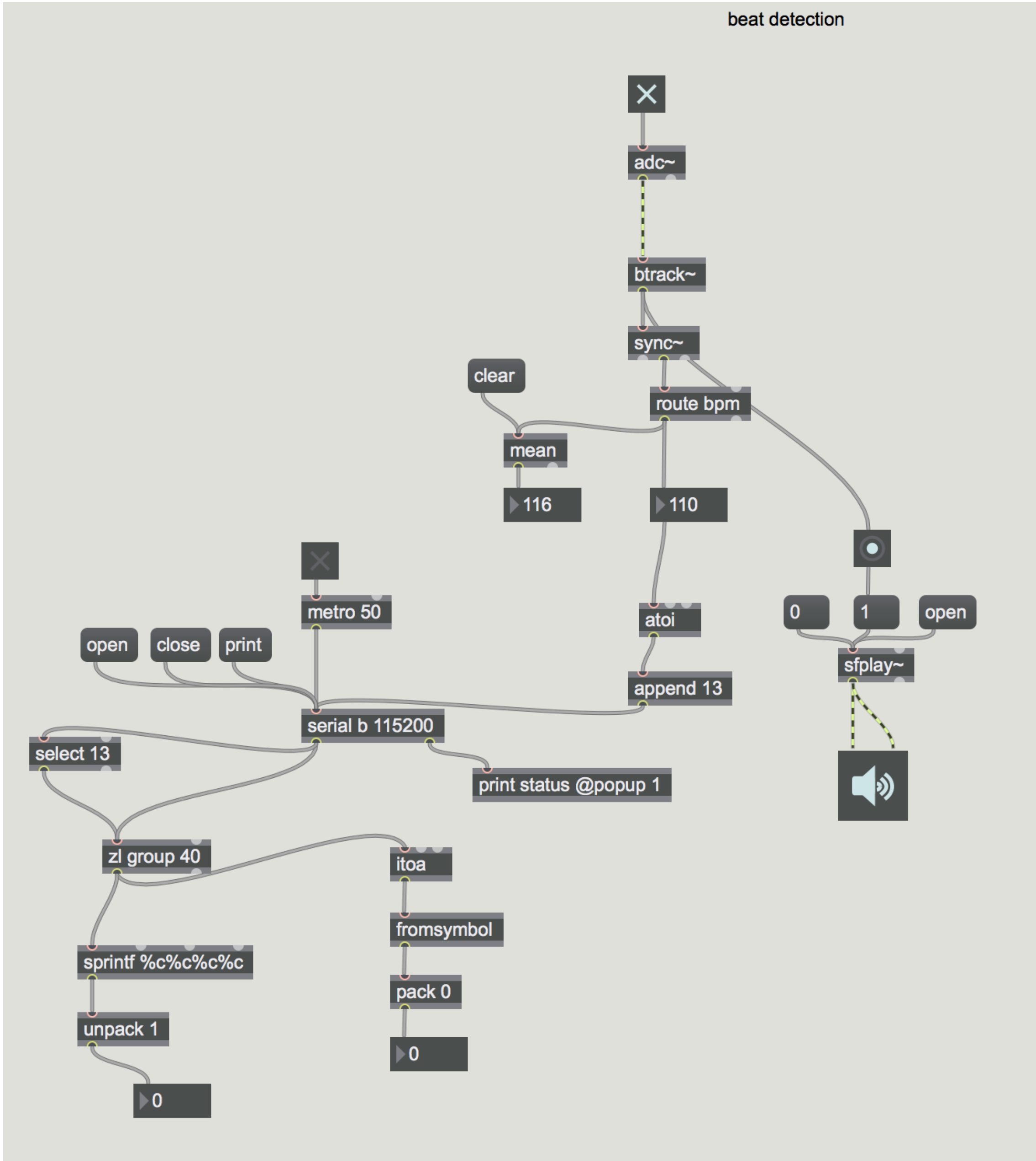


Figure 5.2: Beat tracking max patch with additions for sending bpm over serial

This was an important step to determine feasibility. Though some phasing occurs do to the jittery nature of the *btrack* patch, some intelligent averaging is necessary to lock to a bpm. Work in the future can expand on this conceptually and also provide some windowing or system wide delay to provide further synchronization.

5.1.4 Extra-musical Applications

The same researchers who developed the haptic drumkit discovered the extension of the application towards those with restricted mobility such as stroke patients and those suffering from Parkinson's disease. The results were promising, one patient who was a veteran mentioned that it put a marching sense back into his mind and helped remind him of the sensation of even walking. They morphed the project into what is known now as the haptic bracelet 4 years later [21].

This work has hope to move in the same direction with a system giving constant feedback to inform the wearer of upcoming obstacles and help usher through dynamically changing thresholds.

Appendix A

Haptic Design

This chapter briefly touches on the field of haptics A.1 and delves into considerations impacting design A.1.1. This leads to a discussion of the requirements A.2, initial prototypes A.3, and overall challenges which were overcome that led to the development of the final prototype, the vibrotactile array A.4.

A.1 Brief introduction to haptics

Haptics are the field of research which concern the sense of touch as it applies to *kinesthetic* and *tactile* sensation. The tactile sense enables humans to perceive object properties through skin contact while the *kinesthetic* or *proprioceptive* sense lets one perceive the positions, movements, and forces on one's own body.

The skin is lined with an array of sensory receptors which respond to mechanical pressure and distortions such as skin deformation. The *lamellated* or *pacinian corpuscles* (PC) are responsible for sensitivity to vibration and pressure. These rapidly adapting receptors are responsible for vibrotactile perception in glabrous skin.

Sensitivity to a tactile stimulus grows with the area in contact with the skin and also improves with the stimulus duration until it reaches a point of saturation. When pressure is continuous an effect called *haptic masking* (also known as the *summation effect*) is possible. The overstimulation of the *pacinian corpuscles* causes the brain to ignore these messages with a mechanical filtering system which renders the stimuli to noise in order to focus on other important happenings. If this was not the case, a person could for example feel the pressure exerted by wearing clothing [22]. This phenomena is important to consider when dealing with haptic placement. As mentioned in 2.1.2, when the vibrotactiles were placed over a larger area haptic masking was avoided and the results closely matched the auditory modality.

A.1.1 Haptic Considerations

The following questions arise based on extensive research done by Choi and Kuchenbecker [22] and are crucial concepts underpinning the creation a meaningful haptic:

1. *Can the user feel it?* Perceptibility of vibrotactile stimuli is strongly dependent on the frequency of vibration. The minimum threshold is observed to be between 150-300Hz and can cover an area smaller than 0.1 micrometer. The absolute thresholds are dependent on factors such as body site, contact area, stimulus duration, stimulus waveform, contact force, skin temperature, presence of other masking stimuli, and age.
2. *Can the user distinguish between the different vibrotactile cues being displayed?* This is quantified by the discrimination or *difference threshold* also called the *Just Noticeable Difference* (JND). It is defined as the smallest amount a stimulus intensity much change to produce a noticeable change in sensory experience. The JND is measured as a *Weber fraction*: $\Delta \mathrm{I}/\mathrm{I} = \mathrm{k}$ or the ratio of difference threshold to the reference level. Research into experimental psychology has deemed a 20-30% difference in amplitude or frequency is necessary for robust discrimination between vibrotactile stimuli in practical applications.
3. *How strong does a certain vibrotactile cue feel to the user? Steven's power law* describes the relationship between the magnitude of a physical stimulus and its perceived intensity or strength. See Figure A.1

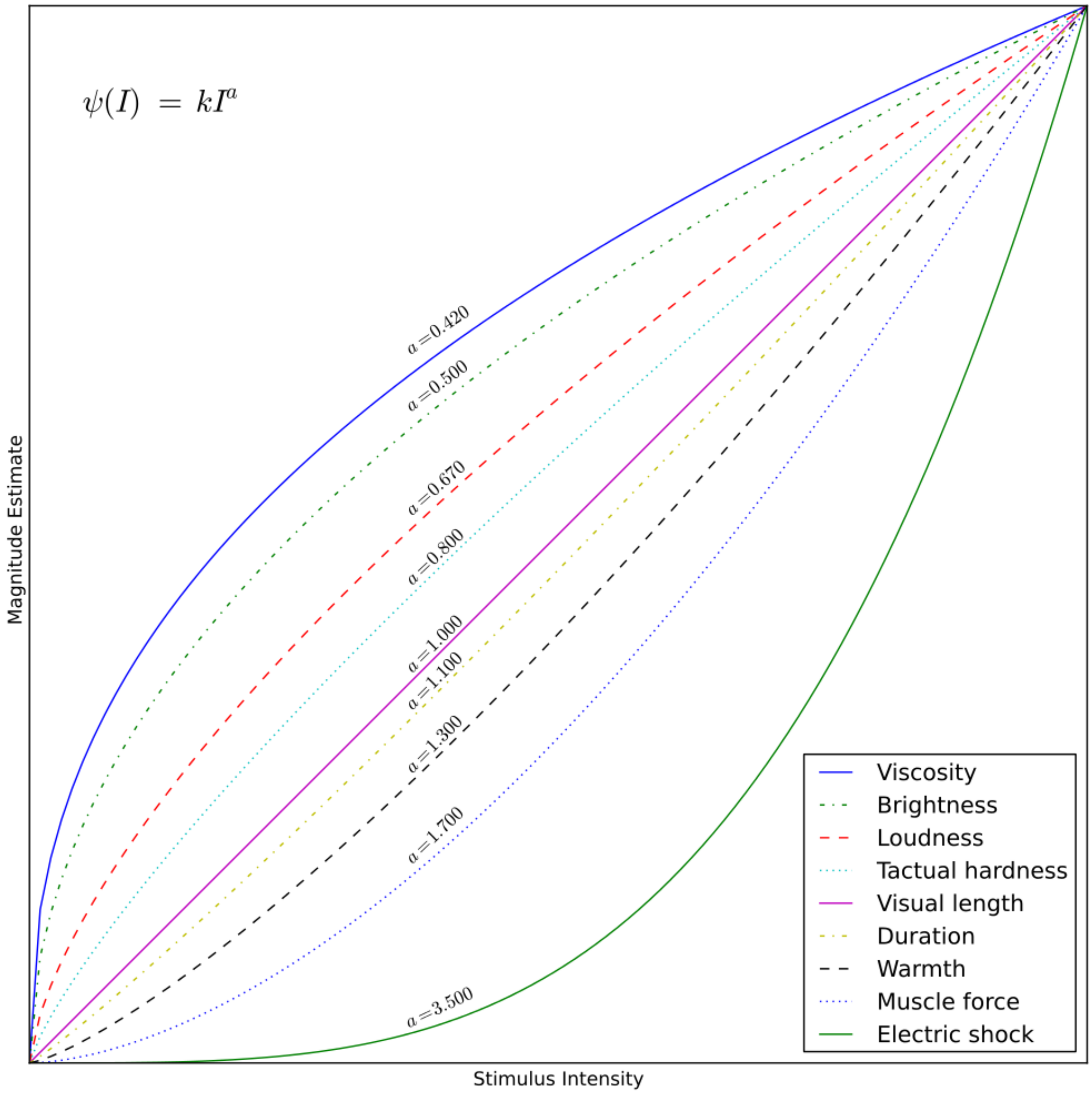


Figure A.1: Steven's Power Law

When a stimulus intensity I is above its absolute threshold, humans perceive its magnitude as $\Psi(I)$(perceptual strength). The exponent (dependent on stimulation freq) determines the growth rate of the perceived magnitude and ranges from 0.35 to 0.86 for vibrotactiles. Perceived intensity is a function of frequency and amplitude of vibration (also affecting perceived pitch).

4. *How good are users at judging timing of vibrotactile cues?* Tactile perception

is generally considered to have high temporal acuity. Vibrotactile temporal resolution research cites a human's ability to distinguish successive pulses with a time gap as small as 5 ms (12000 BPM). This resolution is better than vision (25ms) but slightly worse than physiological experiments into the peripheral auditory system which cites a theoretical best case scenario of approximately 2 ms [23] [24]. Keep in mind that these are temporal resolutions measured with brain scans and do not necessary translate into the limits of sensorimotor response, which, according to prior research, indicates a more realistic resolution of approximately 50ms.

5. *Can Vibrotactile cues elicit any other perceptual effects?* Below 3 Hz is considered slow kinesthetic motion. Between 10-70Hz is the sensation of rough motion or fluttering and between 100-300Hz is the sensation of smooth vibration. Subjective quality of a vibrotactile stimulus can be controlled by modifying the envelope of the stimulus amplitude.

A.1.2 Vibrotactiles

The exploration of touch actuation led to the evaluation of available vibrotactiles. The following is a breakdown of available vibrotactiles conducted to inform design perspective.

1. Linear electromagetic actuators
    - solenoid:
        - can leverage resonance, large output for small input
        - force dependent on position within magnetic field
        - influenced by device orientation relative to gravity
        - heats up during use
    - voice coil:
        - linear dynamics yields consistent output, relatively easy to model
        - *C2 tactor*:
            - 7.6mm contactor preloaded against the skin
            - suspension resonates at 250Hz for maximum perceptibility

    - *Haptuator*:
        - moving magnet design
        - not meant to touch the skin
        - optimized to render frequencies above 50Hz

2. Rotary Electromagnetic Actuators (ERM - eccentric rotating mass)
    - simple, reliable, rotate continuously with a constant voltage/current applied
    - off-center mass affixed to output shaft so that its rotation exerts large radial forces on the body of the motor
    - couples freq and amplitude of the resulting vibration to the motors rotational speed
    - small voltage yields weaker vibrations
    - intrinsic spin-up time could cause delay at the start of the cue
    - internal static friction can prevent motor from rotating when the applied voltage is very small
3. Nonelectromagnetic Actuators - Piezoelectric effect
    - respond to inputs very quickly and can output arbitrary waveforms
    - typically require input on the order of 100V
    - high stiffness of skin creates a need for relatively heavy vibrotactile actuator
    - most don't have power to move the skin without pushing off a cumbersome mechanical ground
4. EAP (electroactive polymer) actuators
    - uses elastomers rather than ceramics
    - can achieve larger deformations for lower drive voltages
5. SMA (shape memory allow) actuators
    - remembers original shape
    - mechanical properties altered in response to temp changes
    - slow response time, large hystoresis, high energy consumption
6. Pneumatic systems

- compact, light
- require high-pressure air source
- struggle to output high-frequency signals

7. Forced impact
   - TacHammer - new technology, specs unknown, hard to acquire

Vibrotactile Constraints

1. Create consistent mechanical coupling between actuator vibrations and users skin
2. Slight changes to such a system drastically affect a users ability to feel and comprehend the rendered signals.
3. For fixed actuator size/activation level, magnitude of created vibrations is inversely proportional to the mass of the object.
4. High bandwidth accelerometer can be used to measure vibration output performance [16].
5. When the application involves a large object, a wearable device, and/or multiple stimulation sites, the optimal vibrotactile rendering paradigm is to vibrate one or more small zones.
   - In a tactile display application the localization accuracy of 250-Hz vibrotactile stimuli around the waist was 74% with 12 equidistant tactile actuators (tactors), 92% with eight tactors, and 97% with six tactors [22].

A.2 Design requirements

The initial requirement set forth by Professor Neely in the Haptic Enviro-Sensing Metronome (HESM) design draft is centered around an analogue wave that could squeeze and release. As the analogue wave approaches its crest it provides insight forecasting the approaching *crusis*, allowing the user to prepare for and rebound from the "click-moment" with rich entrainment.

This observation is in direct parallel to external vibrotactile metronome research as discussed in 2.3. The constraint was such that the pulses should feel continuous and not discrete in order to capture the essence of pendulum motion.

As the intention is to encourage entrainment of the human body to external forces, the frequencies required are quite low, based on the tempi of slow walking to running gaits (40 bpm/.67 Hz to 180 bpm/3 Hz).

A.3 Initial Prototypes

In order to capture the sensation defined in the design requirements, a series of prototypes were rapidly developed.

A.3.1 Solenoid bracelet

Initial introspection towards capturing the squeeze and release sensation led to the rapid prototyping of a simple solenoid bracelet.

Parts List

1. Adafruit Pro Trinket 5V 16MHz
2. N-channel MOSFET
3. 1N4004 diode
4. mini push-pull solenoid

Assembly

The design was inspired and assembled per *Adafruit* specification [25]. The base of an N-channel MOSFET was connected through a 1K resistor to a digital I/O pin on the trinket per Figure A.2. The collector was connected through the solenoid and diode in parallel to Vcc running at 5V.

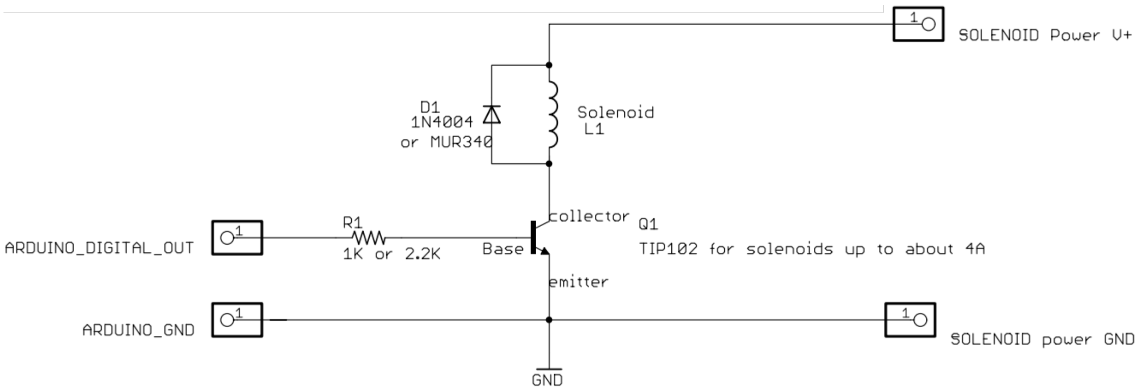


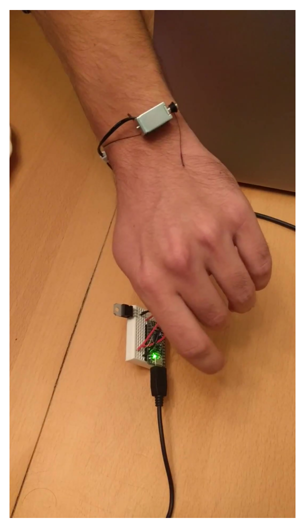
Figure A.2: Solenoid Schematic

Method

As a voltage is applied the slug in the middle of the solenoid is pulled into the center of the coil. The actuation pulls a taught wristband attached to the chasis of the solenoid and as the voltage drops the solenoid retracts releasing tension in the wristband, shown in Figure A.3.

This was controlled in the *Arduino IDE* through a simple PMW signal with increasing duty cycle which output through the digital I/O. The delay was hard coded proportional to the desired BPM.

Figure A.3: Solenoid Wristband Prototype



Outcome

Due to the linear relationship between current draw and pull force, the solenoid required high current and significant voltage thus isolation from the microcontroller was ideal. The necessary rigidity of the band was a cause of discomfort and the lack of positioning options was a detriment to musicians who relied on availability of their hand. Additionally, heat dissipation was at times unsafe and unbearable since the chasis was in direct contact with the skin. Though it captured the tension and release sensation well, there seemed to be a lack of clarity with regards to communication of whether each push pull iteration was a beat length or if a single contraction was the downbeat (i.e. eighth note pulse rather than quarter note). Coupled with the bulky nature of the solenoid chasis, high power requirements and excessive heat dissipation, the solenoid prototype was quickly abandoned.

A.3.2 Single vibrotactile

The subsequent prototype iteration was the first involving a vibrotactile motor. Since the goal was to run everything off of a single board, the voltage constraint was limited to the 5V maximum per *Adafruit Pro Trinket* spec. An ERM motor was chosen for its working voltage range of 2-5V and minimal coin cell form factor (10mm diameter). Like the solenoid, higher applied voltage yielded more current draw but stronger vibration. At 5V, a single motor draws 100mA. The specification was 1100 at 5V which roughly translates to 183Hz. Though not quite at the ideal 250Hz range optimal for skin sensitivity, this was deemed close enough.

To realize the spectrum of capability for vibrotactile sensation (beyond pulse width modulation of the signal) a haptic motor controller with a pre-installed library of effects was acquired.

The goal of this design was to test the ERM sensation on a portable wearable. The MCU was altered from the Pro Trinket to the Flora which ran at 3.3V and had less digital I/O pins, but supported external connectivity and took up less surface area.

Parts List

1. Vibrating mini motor disc
2. Adafruit DRV2605L Haptic Motor Controller
3. Flora Wearable Bluefruit LE Module
4. Flora Wearable electronic platform
5. LiPo Battery - 3.7v 1100mAh

Assembly

First, the ERM leads were soldered to the DRV2605 haptic motor controller and connected via I2C protocol to the complimentary pins on the Flora (SCL,SDA). To experiment with triggering the vibrotactile wirelesssly, the bluetooth low energy (BLE) module was added and the send and receive (Tx/Rx) pins were connected as referenced in Figure A.4. The battery was connected via the built-in terminal clip and last the entire prototype was fitted into the space of a sports wristband with the vibrotactile on the inside touching the skin.

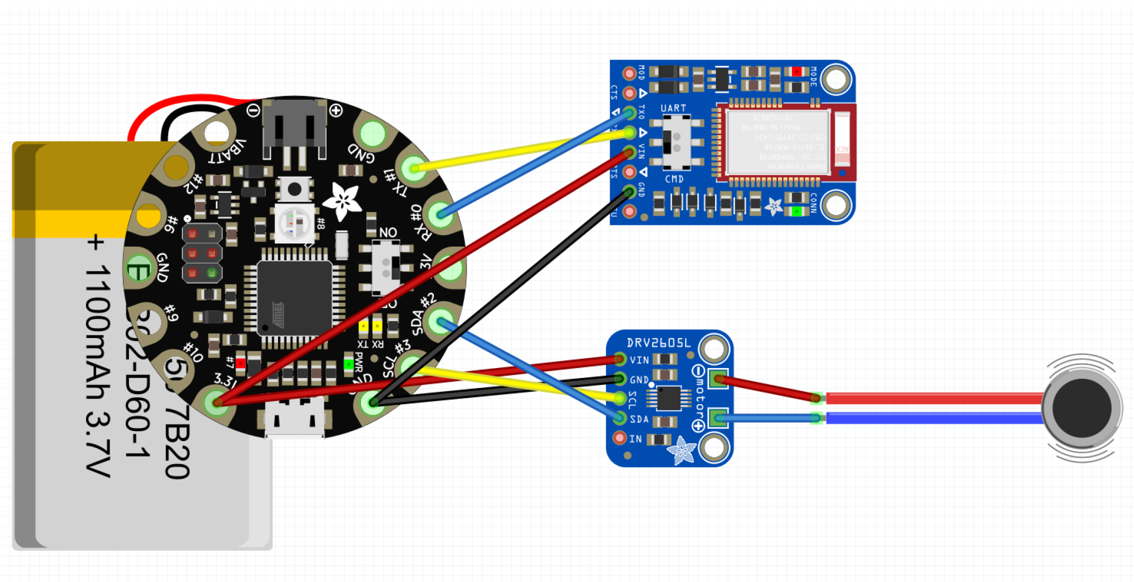


Figure A.4: Prototype 2 - Single vibrotactile, wireless connectivity

Method

Once the hardware was setup, the haptic library was iterated through for selection of the most influential effect. The optimal sensation chosen was a queue of two chained effects according to the DRV2605 datasheet[1]:

- 83 - Transition Ramp Up Long Smooth 2 - 0 to 100%
- 71 - Transition Ramp Down Long Smooth 2 - 100% to 0

[1] http://www.ti.com/lit/ds/symlink/drv2605.pdf

Within the Arduino IDE the Bluefruit library and dependencies were imported and the bluetooth low energy connection configured via UART. On the client side, the connectivity was validated via the publicly available Bluefruit application on an external Android device. The app would send integer values representing the desired BPM to the connected haptic wearable. The code was written such that upon setup and BT pairing, the main loop was polling for packets. Once received in the buffer, the new bpm value was parsed into a period value in milliseconds via equation A.1

$$period = \frac{60,000}{bpm} \tag{A.1}$$

Since the highest operational bpm specified was 180, the shortest period would be an interval of 333.33 ms. This value divided in half gave the maximum allowed ramp up time for the motor, approximately 150ms. The new period value was fed into a state machine which set the on and off state of the motor based on a timer from half the calculated period as well as the 150ms off state.

Outcome

The singular ERM prototype granted key insight into the capability of a vibrotactile to create the desired awareness and fill the interstitial space; though it was found to be lacking the ability to fully command the wearer's attention. This was primarily due to the fact that it was driven by a 3.3V board which inhibited the vibrational strength. The next iteration needed to operate at higher voltage to get a stronger vibration. It was also deemed necessary to increase the number of vibrotactiles to work in an array format in complete synchronicity to explicitly communicate the necessary ramp down and up sensation.

Though the haptic motor controller was a critical evaluation tool for selecting the vibration effect, it was crucial for the final prototype to be able to turn on the motors at full voltage as quick as possible in order to minimize ramp up time. Chaining the motors would also optimize ramp up time in allowing a motor the time to fully spin back down while the adjacent was spinning up.

Furthermore, the *delay()* function added in the BLE section of code was causing the haptic to drift slightly in tempo beyond five minutes of runtime due to the programmatic halting and resumption of dependent timers. The next prototype would have to move away from the delay function which halted program execution.

A.4 Vibrotactile Haptic Array

The final prototype was an array of four vibrotactiles. Several hardware advancements were implemented in order to solve design challenges. The overall process is detailed below.

A.4.1 Hardware

The main board was reverted back to that used in the solenoid prototype, the 5V 16MHz *Adafruit Pro Trinket*, in order to provide maximum possible voltage to the motors. This board did not have built in serial communication so an FTDI to USB cable was necessary in order to communicate with the device. Bluetooth connectivity was abandoned to concentrate focus on minimized latency.

Parts List

Part Type	Properties	Quantity
Electrolytic Capacitor	capacitance 68F; package 0405 [SMD, electrolytic]; voltage 16V	1
Electrolytic Capacitor	capacitance 10F; package 200 mil [THT, electrolytic]; voltage 25V	4
Ceramic Capacitor	capacitance 100nF; package 100 mil [THT, multilayer]; voltage 6.3V	1
Diode	package diode-1n4001; variant pth	4
Vibration Motor	vibration motor 11000 RPM 5VDC	4
Adafruit Pro Trinket 5V 16MHz	variant variant 1; part # Adafruit #2000	1
2N7000 FET N-Channel	package TO92; type n-channel; part # 2N7000	4
220 Resistor	tolerance 5%; package 0805 [SMD]; resistance 220	5
10k Resistor	tolerance 5%; package 0603 [SMD]; resistance 10k	1
FTDI to USB	Adafruit FTDI Serial TTL232 USB Cable [ADA70]	1
Shrink wrap	Heat Shrink Pack	1

Table A.1: Vibrotactile Haptic Array Parts List

Assembly

Since each digital I/O of the Trinket could only source 20mA, four N-channel 2N7000 transistors were chosen to act as switches and current isolators for controlling power to the motors. These are labelled (N) in Figure A.6

Each motor was connected from the power source (5V Vcc), to the drain of the transistor. When the transistor received a signal past its bias voltage of 0.8V it was switched on. This allowed the drain-source channel to be opened and an onrush of current to flow from Vcc to the motors and through to ground.

A 1N4001 diode was placed in parallel with the motor from the 5V Vcc node to the drain of the N-channel 2N7000 to protect the transistor by shorting out the onrush of back current emitted from the motor during immediate shutoff. Principally, the motor will act as an inductor; a sudden change in current creates an equivalent voltage to keep that current flowing short term. This could fry the transistor if the diode was not in place to short out this negative voltage spike as shown in A.5

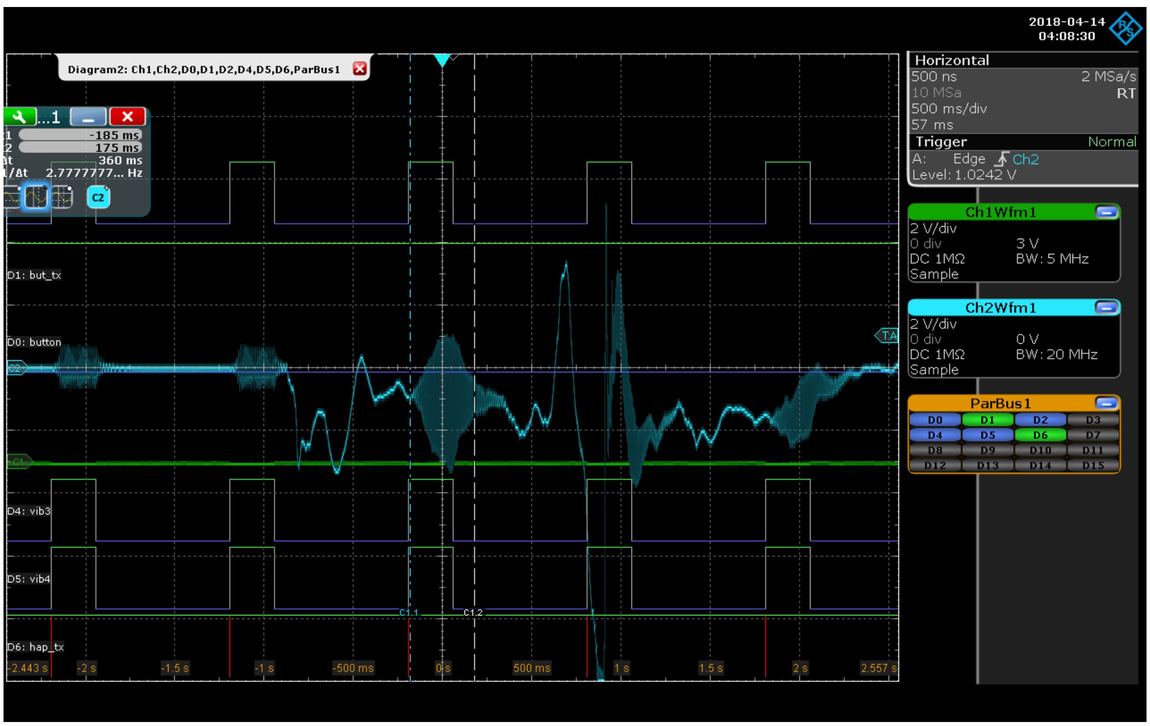


Figure A.5: Motor ringing after abrupt shutoff

The digital pins 5,6,9,10 (denoted with yellow and grey wires in Figure A.6) output a pulse-width modulated signal to be used as triggers, or control switches, for the transistors. In order to limit the current that the digital output must source and to protect the transistor gate, a 220 Ohm resistor was connected across the digital I/O pin and the gate of the 2N7000 for each pin.

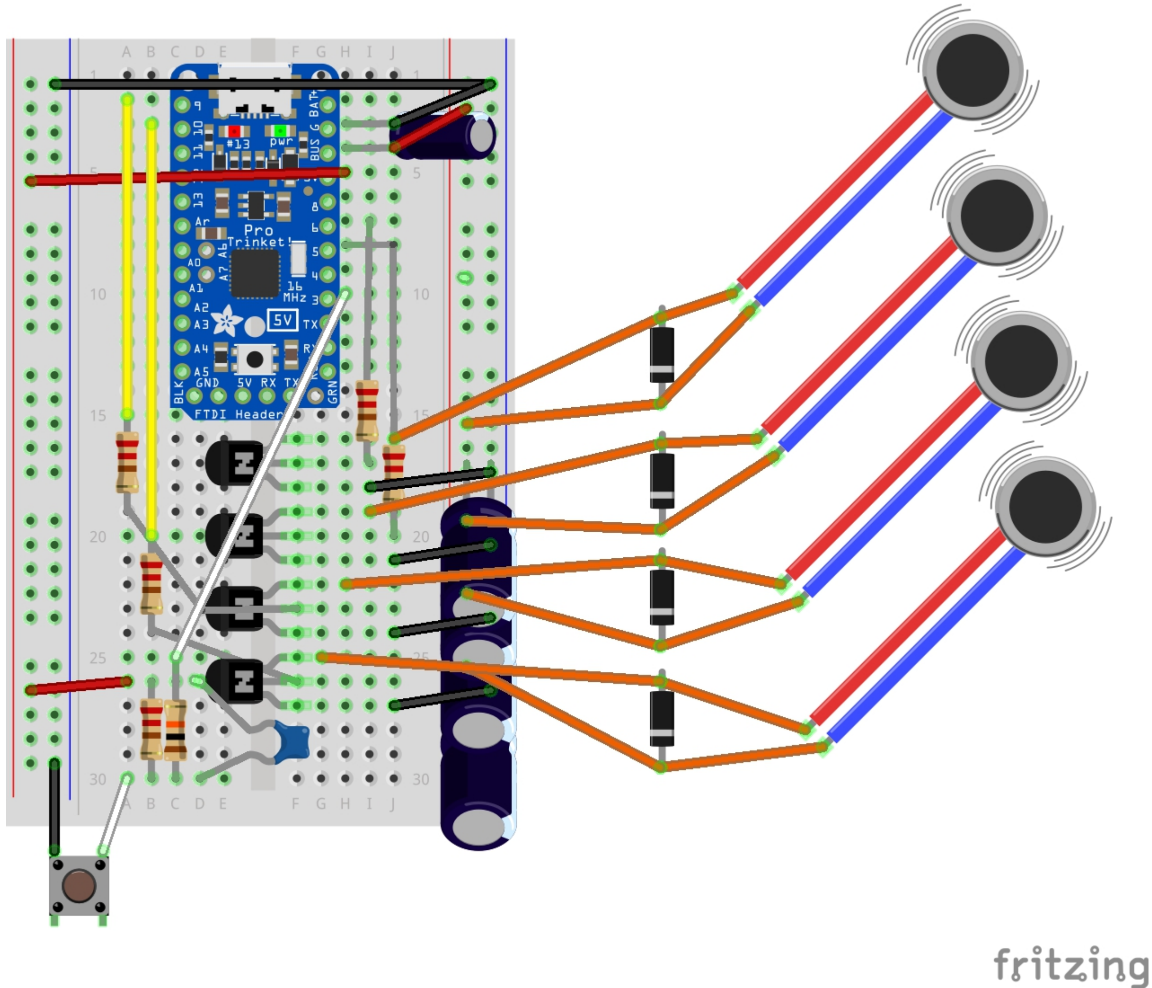


Figure A.6: Final prototype wiring mockup

The voltage across the rails (Vcc and ground) was viewed on a Rohde & Schwarz RTO 2004 oscilloscope. The analyzed waveform showed some unfavorable dips primarily when all motors were running due to the high current draw from the ERMs in addition to some ringing (overshoot), seen in Fig A.7.

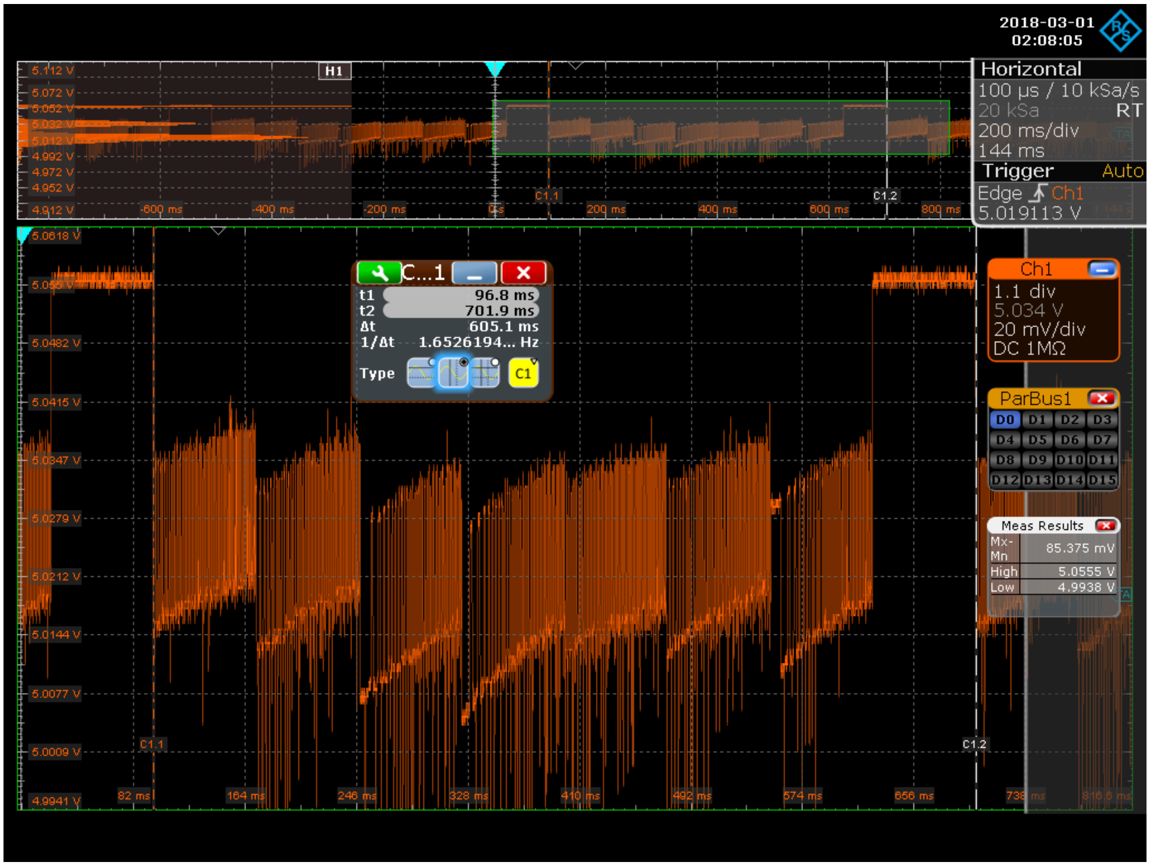


Figure A.7: V drop across Vcc and HF ringing: Pre-cap

A.4.2 Improvements

Capacitors were added to act as buffers from the power source to the motors, in doing so they helped provide immediate current to the motors when the PWM input signal engaged the transistor and the motor would go from an off state to an immediate on state drawing high amounts of current. Large electrolytic capacitors are known for their ability to supply high currents for a few milliseconds, more so than a battery or in the haptic wearables case, USB power. These were added across Vcc and ground nearest to the Trinket as well as across each of the node rails nearest to the motors. The change in output shape can be analyzed in Fig A.8.

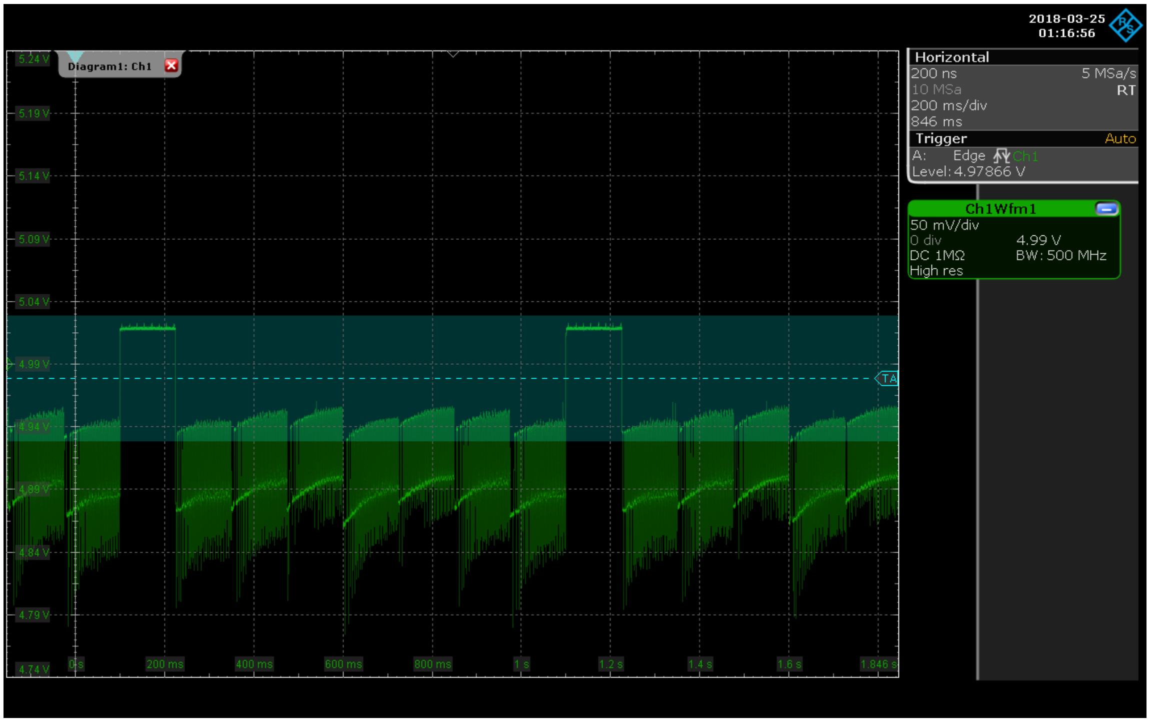


Figure A.8: V drop across Vcc: Post-cap

As an additional manual tap tempo option for the user to experiment with, a push-button was added and connected to the only interrupt capable pin on the Trinket, PIN 3. An RC combination was chosen to act as a low-pass filter to protect against debounce scenarios.

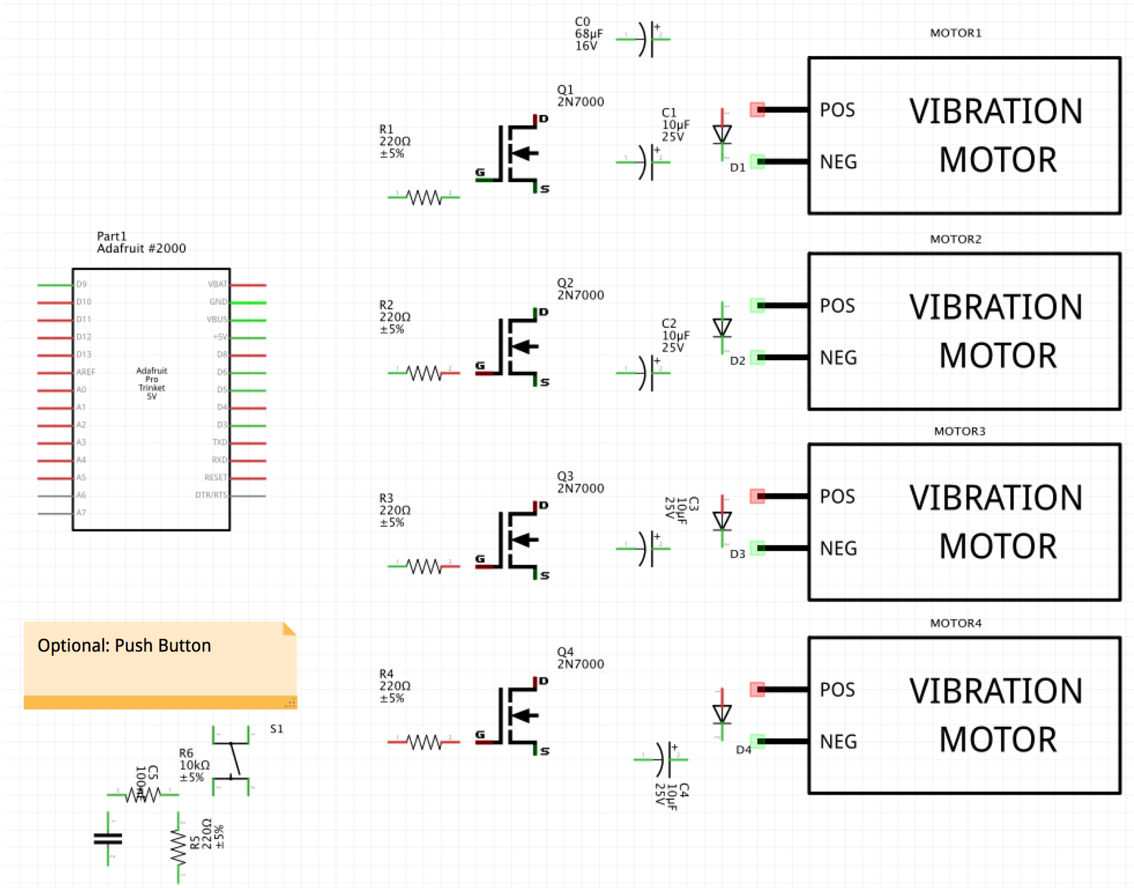


Figure A.9: Final prototype abstracted schematic

Motor Characterization

In order to determine an accurate baseline for motor ramp up, a singular motor was firmly attached to a piezoelectric transducer. The remaining motors were connected to the digital bus of the scope for timing analysis.

Figure A.10: Motor ramp up time

As seen in Fig A.10, the motors reach full amplitude over approximately 67 ms before the signal goes low and starts to decay. The AC shape of the waveform is due to the nature of the ERM. The rotating mass translates left and right movement into a voltage oscillating in amplitude. Multiple ramp up times were averaged using this method such that the time before perceptibility was determined to be about a quarter of the ramp up time (approximately 50ms), in agreement with 2.2. This is clearly shown in Fig A.11 below.

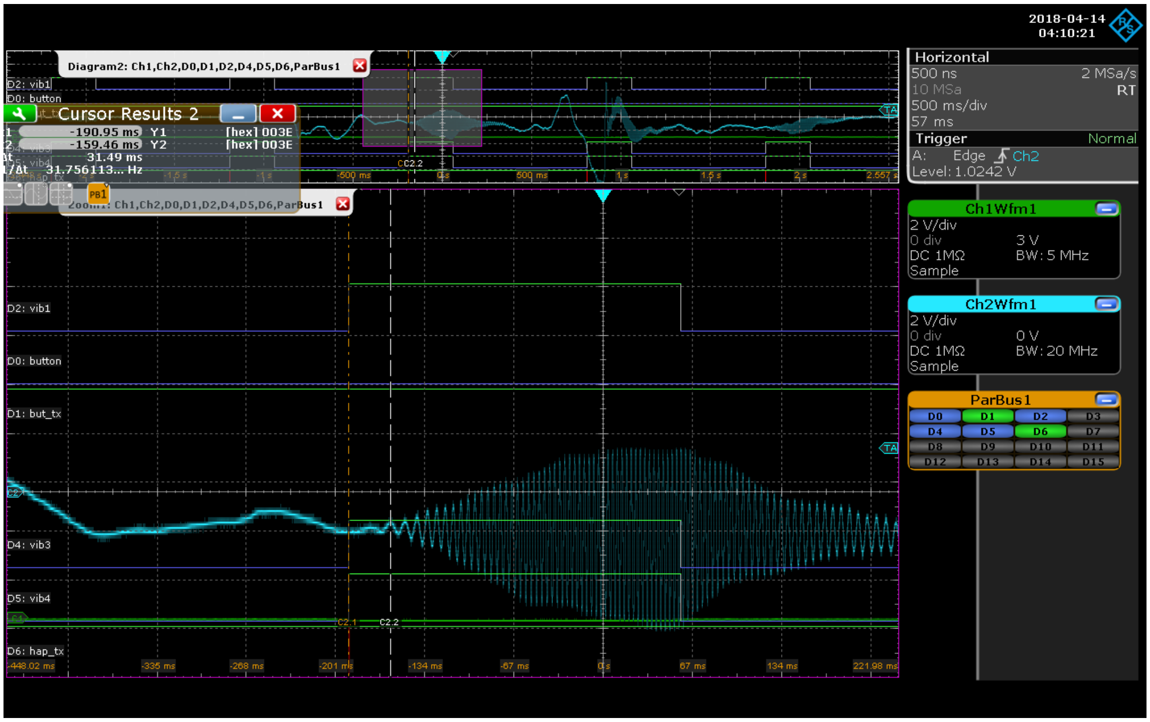


Figure A.11: 50ms motor ramp up time to perceptibility

A.4.3 Haptic Software

The code was written in the Arduino IDE which utilizes C++. Outside of the tap tempo hardware push button, the main control flow was through serial communication over FTDI. The device communicated at 115200 baud and a small method was written to read the serial buffer and parse the incoming bytes into integers. The first input was a setter to store the current operation mode with 1 being the Discrete (Instantaneous) Mode and 2 Continuous. Afterwards, any other number (thresholded from 20 to 220) would be stored as the bpm. The period was calculated using formula A.1

The main control flow was two state machines which were delay independent. Depending on mode selection it would send digitalWrite commands to each motor within the set time period. Though the analogWrite functionality built into the Arduino IDE could PWM the signal and potentially control the motor just below its turn on state, the transistors would not allow this due to their rapid speed and turn on voltage and the motors would be always vibrating. This was a design decision informed by the previous single vibrotactile prototype implementation.

The first state started a millisecond precision timer. In discrete mode the second state wrote all digitalIO pins HIGH for a quarter of the period. A half on/off implementation was considered but determined to be not as efficient in communicating rapid pulses since the ramp down time overlapped with the next on period. During this state a flag was printed over serial to be read in later indicating when the test software could timestamp the onset.

```
void go(){
    if(discrete==true){
        switch (newState){
          case READY:
            if(start){
              average = avg;
              startTime=millis();
              newState++;
            }
            break;
          case ON:
            digitalWrite(vibPins[0],HIGH);
            digitalWrite(vibPins[1],HIGH);
            digitalWrite(vibPins[2],HIGH);
            digitalWrite(vibPins[3],HIGH);
            if((millis()-startTime) >= ((average * 1)/4)){
              newState++;
              Serial.println("onset");
            }
            break;
          case OFF:
            digitalWrite(vibPins[0],LOW);
            digitalWrite(vibPins[1],LOW);
            digitalWrite(vibPins[2],LOW);
            digitalWrite(vibPins[3],LOW);
            if(millis()-startTime >= average){
              newState=0;
```

```
                }
                break;
            default:
                break;
        }
    }
```

The state machine in continuous mode functioned similarly but was divided into 4 ramp-up states and 4 ramp-down states. Each state held its vibrotactile high for 1/9 of the overall period or IOI but lingered on the fourth vibrotactile slightly longer *(2/9th's * IOI)* to convey the pinnacle of the beat. This state also sent the onset trigger.

```
if(discrete==false){
    switch (state) {
        case START:
            if(start){
                average = avg;
                startTime=millis();
                state++;
            }
            break;
        case RAMPUP_STEP_1:
            digitalWrite(vibPins[0],HIGH);
            if((millis()-startTime) >= ((average * 1)/9)){
                state++;
            }
            break;
        case RAMPUP_STEP_2:
            digitalWrite(vibPins[0],LOW);
            digitalWrite(vibPins[1],HIGH);
            if((millis()-startTime) >= ((average * 2)/9)){
                state++;
            }
```

```
      break;
    case RAMPUP_STEP_3:
      digitalWrite(vibPins[1],LOW);
      digitalWrite(vibPins[2],HIGH);
      if((millis()-startTime) >= ((average * 3)/9)){
        state++;
      }
      break;
    case RAMPUP_STEP_4:
      digitalWrite(vibPins[2],LOW);
      digitalWrite(vibPins[3],HIGH);
      if((millis()-startTime) >= ((average * 5)/9)){
        state++;
        Serial.println("onset");
      }
      break;
    case RAMPDOWN_STEP_5:
      digitalWrite(vibPins[3],LOW);
      digitalWrite(vibPins[2],HIGH);
      if((millis()-startTime) >= ((average * 6)/9)){
        state++;
      }
      break;
    case RAMPDOWN_STEP_6:
      digitalWrite(vibPins[2],LOW);
      digitalWrite(vibPins[1],HIGH);
      if((millis()-startTime) >= ((average * 7)/9)){
        state++;
      }
      break;
    case RAMPDOWN_STEP_7:
      digitalWrite(vibPins[1],LOW);
       digitalWrite(vibPins[0],HIGH);
```

```
        if((millis()-startTime) >= ((average * 8)/9)){
          state++;
        }
        break;
      case END:
          digitalWrite(vibPins[0],LOW);
         if((millis()-startTime) >= average){
           state=0;
         }
        break;
    default:
        break;
    }
}
```

To confirm the period of each motor the transistor gates were connected to a USB logic analyzer measured at 60 bpm (1000 ms). The behavior was precisely as expected as shown in Figure A.12

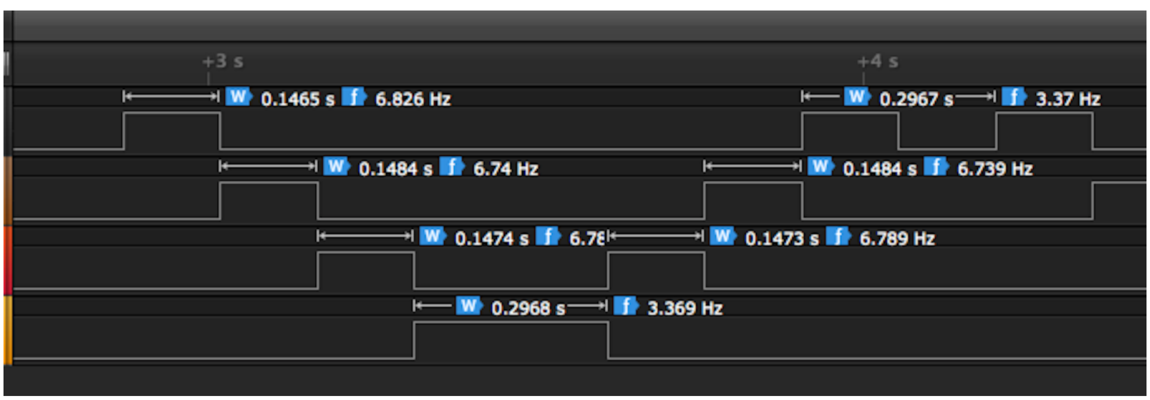


Figure A.12: Period confirmation from gates for ramp up and down mode

Appendix B

Test Suite

B.1 Software Development

A pseudocode breakdown of the framework is provided below. The methods were written in object oriented fashion. The critical sub-components of the test suite are discussed below. Please note, it would be most beneficial to follow along with *testSuite.py* from the repository[1].

B.1.1 GUI

The GUI was written with the Tk framework[2]. It consisted of three frames and a main class which instantiated the others. The *StartPage* stored the *userName* to be later anonymized via a simple char to int summary conversion technique. Upon clicking, the user was presented with the *InstructionPage*. Once the terms of the test were understood and accepted, the final *TestPage* prepares the user for the upcoming practice tests. The main code execution triggers once the **Start Test** button is pressed.

[1] https://github.com/afaintillusion/he-sm
[2] https://en.wikipedia.org/wiki/Tk_(software)

B.1.2 Multithreading

Since the test suite read from two serial devices simultaneously, there needed to be a non-blocking methodology that allowed uninterrupted flow control; this was provided from the import of *threading* from the Python *Thread* library. Once the user started the test, the practice mode began. Here a thread was initiated which created the beep timer. At the end of the audible tone the thread completed and a second thread started which passed the arguments of the test cases (either Haptic or Audio) into the method. A third thread started simultaneously as the second and focussed on acquiring the tap onsets. These threads worked in parallel during the test case duration and upon completion passed arguments to the data analysis method for logging in a continuously appending Pandas dataframe.

B.1.3 Haptic Onset Detection

The haptic test cases were broken down into dictionary lists containing arguments to be passed to the *haptic* method. The arguments were dependent on the test case mode of operation, desired tempo, execution time, and whether the tempo was to change dynamically over time.

The mode and tempo are written over serial to the haptic device and a start time is recorded. From here the program enters a while loop for the duration of test execution reading actively from serial to await an *onset* message. Once received it appends to a list for future data logging.

If however, the test case calls for the dynamic mode of operation, the execution time is broken into quarters. Each quarter either increments or decrements the tempo based on the argument passed in by the test case. In this fashion, the dynamic haptic test cases closely emulate the sinusoidal tempo automation of the dynamic audio tests.

B.1.4 Tap Onset Detection

The tap method read from the incoming Arduino Uno serial line and decoded bytes from the buffer. The preamble was the char B which signified an incoming packet followed by the binary data sequence and terminated with an E. The onset was timestamped and stored in a dataframe and this process repeated until either the haptic or audio methods passed through the closeFile global variable signifying the end of execution.

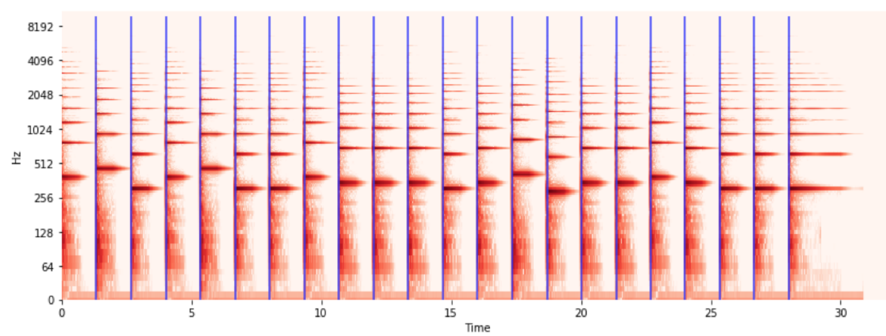
B.1.5 Audio Onset Detection

The wav files were individually analyzed using the *librosa* package based on Steve Tjoa's MIR (Music Information Retrieval) website.[3] The onsets were detected via librosa's onsetDetect method which computed the spectral novelty function to find the peaks. The computation was then converted from frames to seconds with nearly nanosecond precision shown in Figure B.1. The onsets for each audio test case were then stored in dictionary lists to be called upon later by the test suite in order to add timestamps to the seconds stored in the list.

[3] https://musicinformationretrieval.com/onset_detection.html

```
In [28]: onset_times = librosa.frames_to_time(onset_frames)
         print(onset_times)

[ 1.34675737  2.69351474  4.01705215  5.34058957  6.68734694  8.01088435
  9.35764172 10.68117914 12.02793651 13.35147392 14.67501134 16.02176871
 17.34530612 18.69206349 20.01560091 21.36235828 22.68589569 24.00943311
 25.35619048 26.67972789 28.02648526]
```

```
In [29]: S = librosa.stft(x)
         logS = librosa.amplitude_to_db(abs(S))
         plt.figure(figsize=(14, 5))
         librosa.display.specshow(logS, sr=sr, x_axis='time', y_axis='log', cmap='Reds')
         plt.vlines(onset_times, 0, 10000, color='#3333FF')

Out[29]: <matplotlib.collections.LineCollection at 0x114e72e80>
```




```
In [30]: plt.figure(figsize=(14, 5))
         librosa.display.waveplot(x, sr=sr)
         plt.vlines(onset_times, -0.8, 0.79, color='r', alpha=0.8)

Out[30]: <matplotlib.collections.LineCollection at 0x1140e7a58>
```

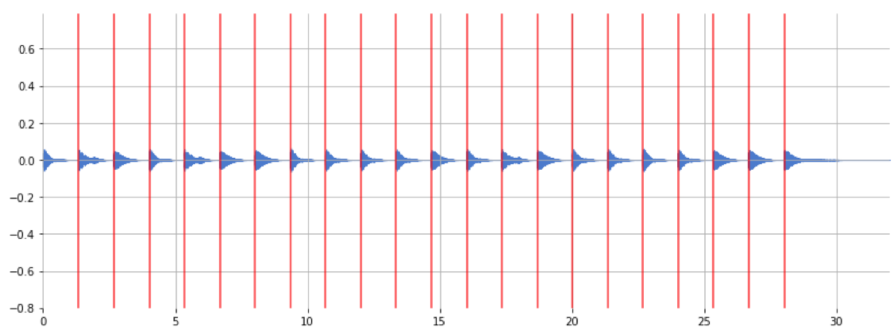


Figure B.1: Test Case A2a1 Audio Onset Detection Example

B.1.6 Audio Rendering

A mixer object was imported from *pyGame*[4], a module for loading sound objects and controlling playback. The object was initialized at 16 bit 44.1 kHz over 2 channels with a buffer size of 64. The buffer size was experimented with to balance lowest latency without audio dropout. The playback method loaded the audio file passed through the method argument and set the volume. Once the mixer play method was called a separate *getBusy* object ran in a while loop during audio playback to find out if the buffer was still busy or not. This object appended the audio file playback start time to a temporary list which was added to the audio onset dictionary list aforementioned and written to a dataframe.

B.1.7 Data Logging

All of the timestamps and onsets were piped into a Pandas[5] dataframe. The dataframe works almost like an SQL table, allowing for mutable size, different column types, manipulation of axes, arithmetic operations, and easy plotting. Duplicate entries recorded were dropped for same measure along with the first and last values of each test to ensure fairness. From the tap and true onsets, the asynchrony was calculated and stored in milliseconds. From the difference between the current line and next line true onset, the inter onset interval (IOI) was recorded and output in milliseconds. Missed taps were tallied based on null values for tap onsets where true onsets existed. Last, the phase correction response was gleaned as the delta between current and next asynchrony.

B.1.8 Sanitization Procedure

It was important to have a method in place for synchronizing results if the user ever missed a tap and tried to get back on beat. Without it, there would be an undesired shift in the dataframe and the asynchrony values would be both miscalculated and unrealistically high. The numpy select method was utilized to filter choices based on conditions within an array (or dataframe). A maximum point of acceptance was

[4] https://www.pygame.org/
[5] https://pandas.pydata.org/

established by taking half the IOI added to the true onset. Similarly the minimum value was taken as half the prior IOI added to the true onset. Every tap onset within a test cases dataframe was shifted through the max and min thresholds to determine whether a tap onset belonged to a different true onset. The new column created was called the *Sanitized Tap Onset* and from here the *Sanitized Asynchrony* was calculated.

B.1.9 Plotting

At the end of the data analysis method a plot based on the current dataframe test case was performed using plotly[6]. This was a snapshot of the time versus onsets for each test case and can be seen in Figure 4.4.

B.2 Tap Test Hardware

Closely following the design outlined in [8], a force sensitive resistor was connected across 5 V and the analog input pin A0 of the Arduino Uno. Bridged between A0 and GND was a 10 K resistor acting as a resistive divider. The code was based on the *fsrSilentDisc.ino* sketch developed by Schultz but expanded and modified to better suit this project.

[6] https://plot.ly/

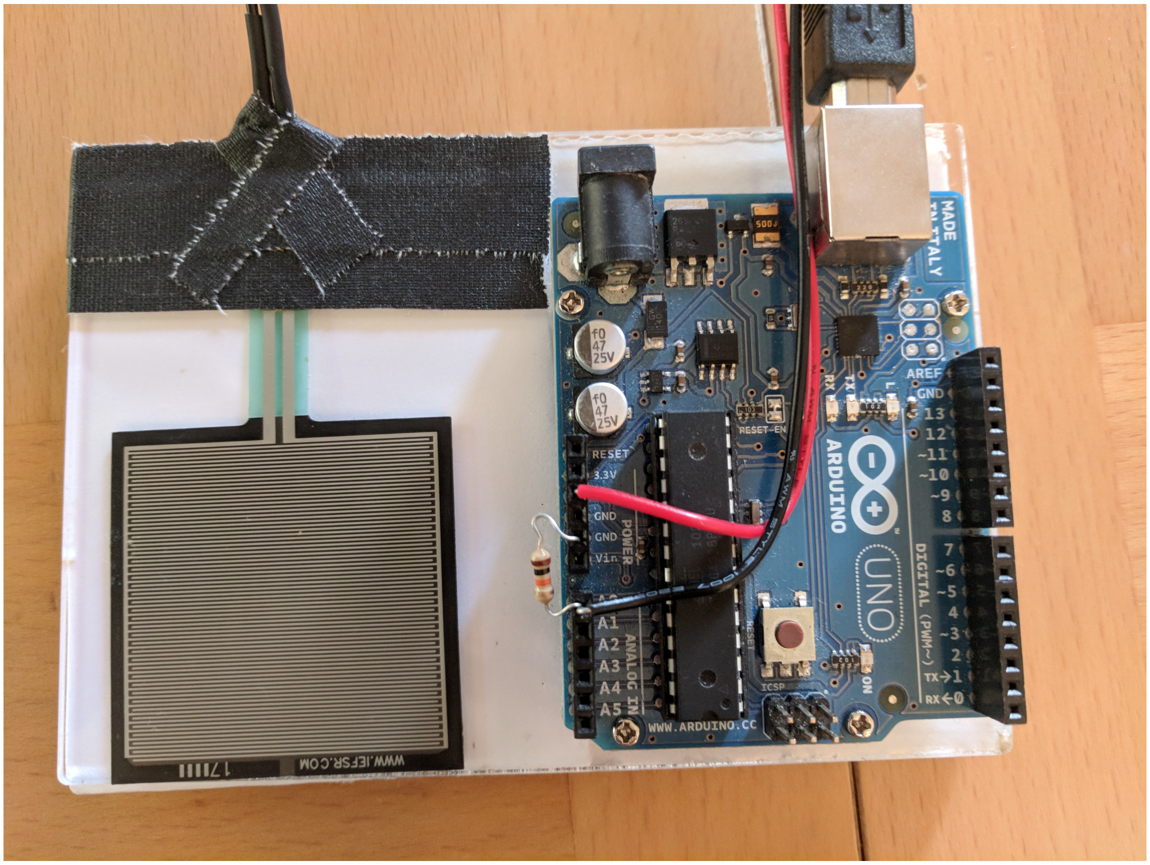


Figure B.2: Arduino Tap Test Hardware

Schultz had defined thresholds to determine the minimum FSR tap necessary to classify as a response. If the analog reading exceeded the threshold then the tap was windowed further to prevent debounce and the data packet readied for transmission. The packet consisted of the character ”B” followed by onset time, offset time, and max force from the FSR reading, ending with the character ”E” to signify the end of the packet. The trade off between responsiveness and threshold required to prevent a double bounce was finely tuned via oscilloscope and will be further explained in Section B.3.

B.3 Latency Evaluation

Significant efforts were made to determine the overall latency of the test system and instill a level of confidence in the validity and accuracy of the data points acquired. The sources of potential latency were isolated into the components identified below.

- FTDI-USB communication from Pro Trinket (16 MHz) to laptop
  - According to section 3.1 of the datasheet for FTDI based chipsets[7] , FTDI USB-Serial communication to PC exhibits, by default, a round trip delay of 16 ms intrinsic to the packet scheduler. When the latency timer expires and the buffer is not yet full any data in the 62 byte buffer is sent along with a short 2 byte status message (total of 64 bytes).
  - This 16 ms spec was confirmed by measuring incoming "onset" messages from the Pro Trinket over FTDI and comparing it to the known period. The jitter measurement was calculated and averaged via a custom script written in *Go* and was found to average 8 ms in either direction, or a span of 16 ms. The swing seemed dependent on the period but always averaged to around 16 ms. See Figure B.3 for confirmation.

```
# Starter: 2018-07-18 14:44:43.213670967 -0400 EDT m=+78.806387300
Diff: -4.172615ms
# Read 7 bytes
# Starter: 2018-07-18 14:44:43.725504457 -0400 EDT m=+79.318220790
Diff: 11.83349ms
# Read 7 bytes
# Starter: 2018-07-18 14:44:44.221297223 -0400 EDT m=+79.814053556
Diff: -4.167234ms
```

Figure B.3: 16 ms jitter on average for single device drift between "onset" message across FTDI over a set period of 500 ms

  - Serial round trip latency (TX Jitter)
    - The round trip latency of a single packet containing the "onset" trigger was sent across a digital pin on the Trinket over to a digital pin on the

[7] http://www.ftdichip.com/Support/Documents/AppNotes/AN232B-04_DataLatencyFlow.pdf

Arduino Uno, which triggered it's own serial "onset" trigger message.

 – The implications of this isolate the USB protocol scheduler as the main culprit for the latency seen during the test.

- The time to actually send the "onset" message across serial was a negligible 0.6 ms as shown in Figure B.4.

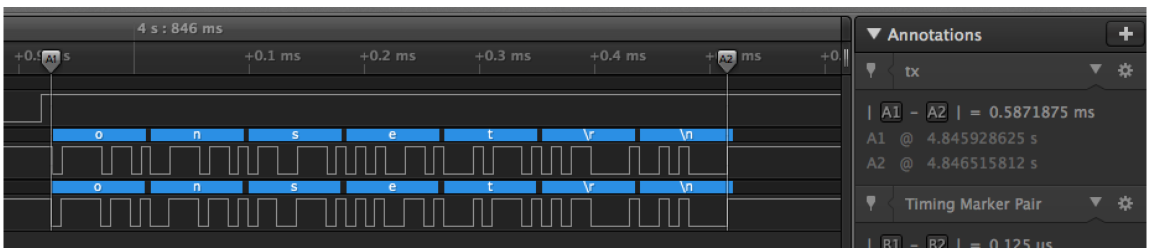


Figure B.4: From the top: 1000 ms starter signal from Pro Trinket. Middle: starter TX line, End: stopper TX line on Arduino Uno

- Time from FSR tap registry to Arduino TX
  - Human input is never perfect. The tap onset needed to be triggered above the noise floor just enough to get the input. It took a few optimization runs to determine what constituted a real tap without triggering a second tap unintentionally. The original *Tap Arduino* code would send the TX message only during the FSR ramp down or decay which introduced some latency. This bound was modified to instead be placed at the ramp up which minimized latency from tap to transmit to 294.4 microseconds as shown in Figure B.5

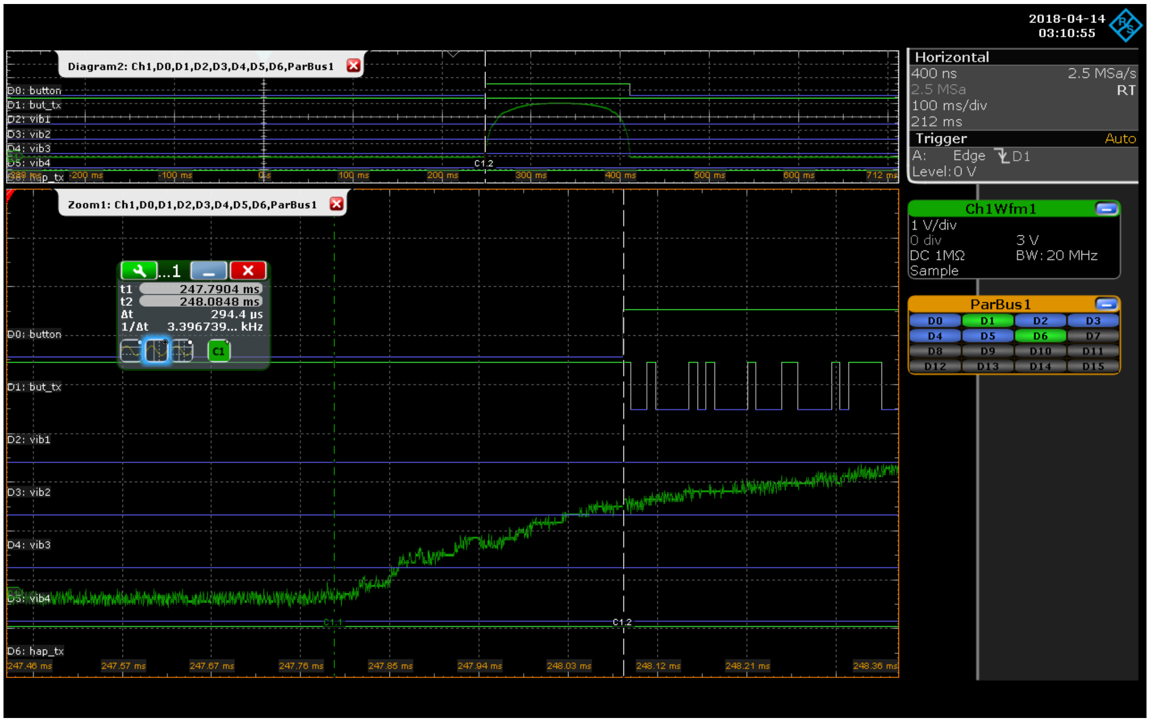


Figure B.5: FSR Button ramp up time before TX: 294.4 microseconds

- Haptic Device Latency
  - In order to quantify latency within the haptic device tests were repeatedly run within a closed loop environment. This meant that the onset trigger I/O pin at the gate node of the Pro Trinket was connected to the A0 pin on the Arduino Uno, emulating the same effect as a tap of the FSR.
  - The haptic test cases were run and the overall average asynchrony between true onset and tap onset across all test cases was found to be approximately -5 ms. The average on an individual test case basis would later be added from the user test results to ensure fair results.
  - A correction factor was necessary since the gate would open and trigger the tap before the onset at the end of motor 4. This time window changed based on the period and mode of operation.

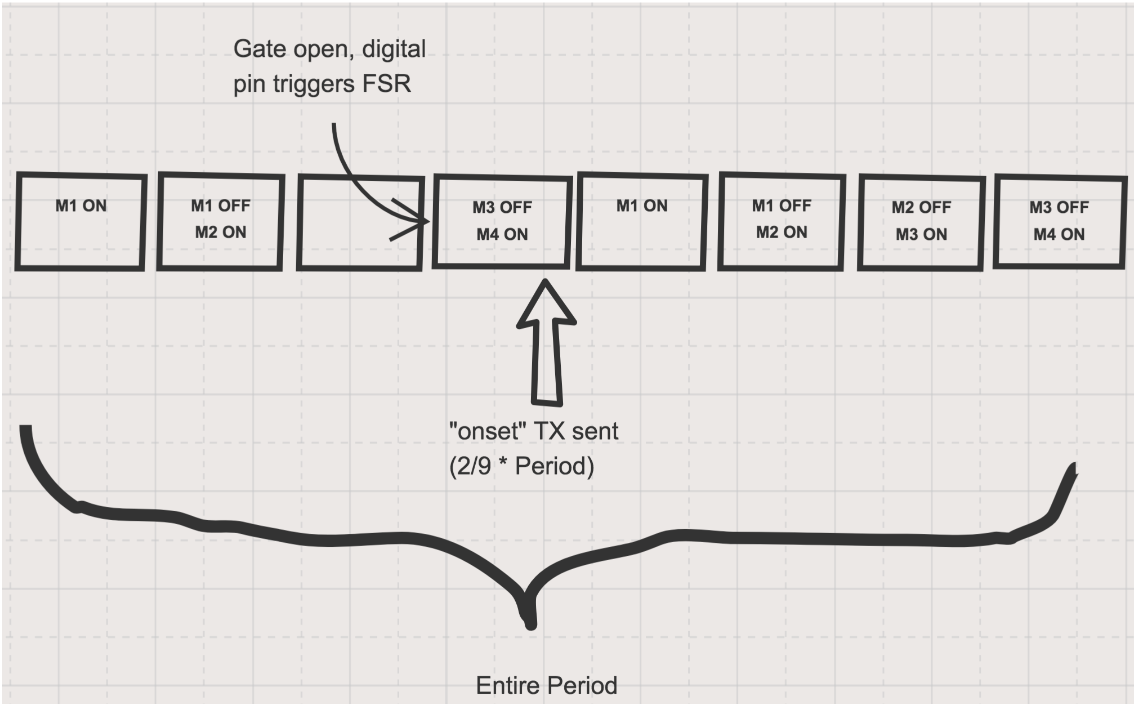


Figure B.6: Diagram of gate versus actual onset

  - Onset trigger in continuous mode 2/9ths (5/9-3/9) of the period. For a 1000 ms period the onset transmission does not occur until 2/9 * 1000 = 222.222 ms as confirmed in Figure B.7.

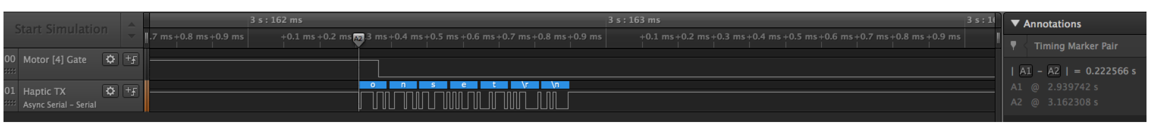


Figure B.7: Haptic correction factor based on gate opening vs. actual onset message

  - The onset trigger in discrete mode starts 1/4 of a period after.

- Audio Latency
  - Audio driver and sound card latency estimate of laptop[8]
    - Tested via loopback method[9] with audio loopback dongle[10] at 44.1 kHz

[8] Mid 2012 Macbook Pro Retina 16GB RAM 2.6GHz i7
[9] https://manual.audacityteam.org/man/latency_test.html
[10] https://source.android.com/devices/audio/latency/loopback

24 bit with an I/O buffer size of 64 samples. The resultant roundtrip latency was a negligible 3.4 ms.

- Test system/Algorithmic latency (pyGame Audio library)
  - A sound detector circuit was used which consisted of a electret microphone, preamp, envelope follower, and Schmitt trigger. The microphone sensitivity was calibrated to match the headphone input. The laptop output was shifted to a single channel such that it was only outputting from the left to avoid any time delay introduction from the right.

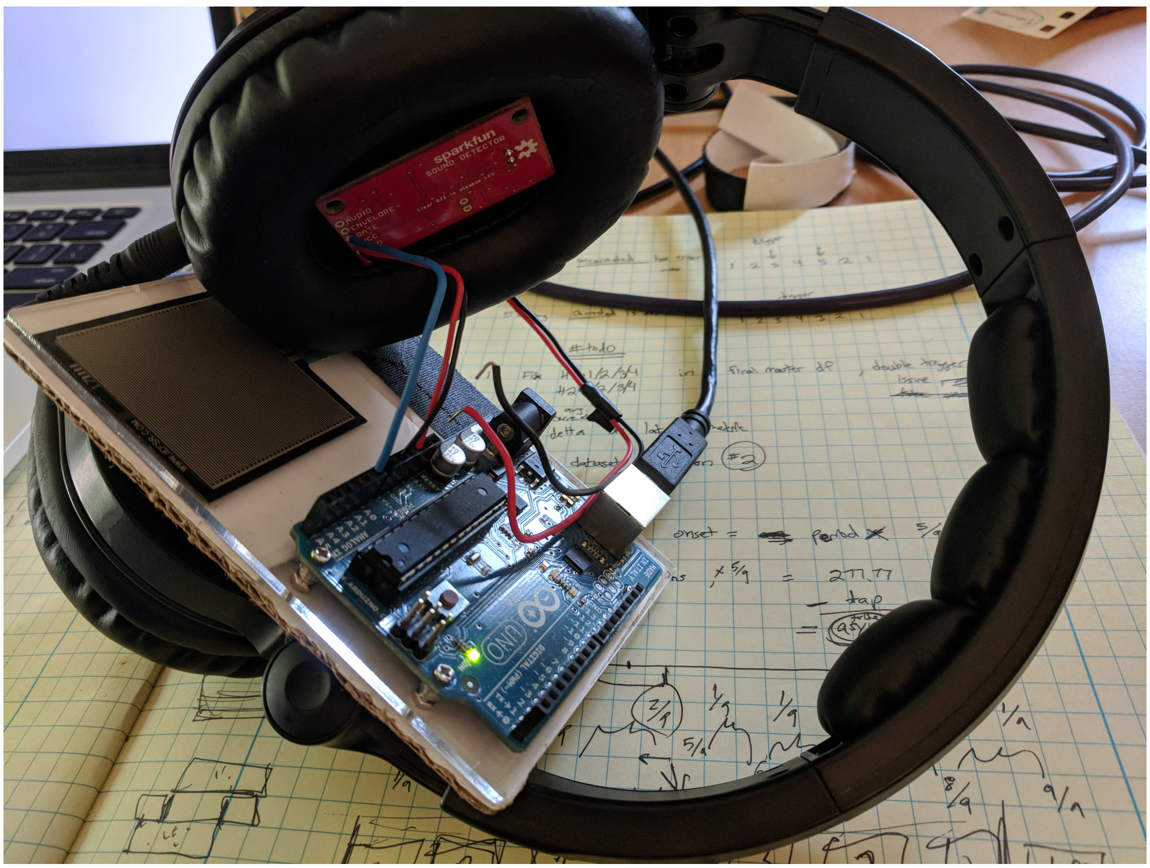


Figure B.8: Audio Latency Setup

  - The gate pin of the sound detector was connected directly to the A0 input on the Arduino Uno. Any impulse exhibited by the headphones triggered the gate to output a digital signal. This triggered the Arduino identically to how the FSR registered a tap.

- The audio test cases were run 5 times each. The discrepancy between true onset and tap onset represented the audible delay in the signal chain. This was averaged across all tests to equate to approximately 32.64 ms.

B.3.1 Closed loop adjustments

Although it was important to classify the magnitude of delay introduced by two simultaneous serial devices, the closed loop haptic and audible tests were the most crucial to compensate for which meant correction on an averaged per test case basis. The overall for audible tests equated to an average reduction of 32.64 ms and for the haptic based tests an addition of 5 ms.

Bibliography

[1] E. C. Fitts, *The New Music Review and Church Music Review, 1901-1935: Keyword-author index: A capella-English text*, vol. 3. Nisc, 2008. 1.1

[2] M. Clayton, R. Sager, and U. Will, "In time with the music: the concept of entrainment and its significance for ethnomusicology.," in *European meetings in ethnomusicology.*, vol. 11, pp. 1–82, Romanian Society for Ethnomusicology, 2005. 1.2

[3] S. Holland, A. J. Bouwer, M. Dalgelish, and T. M. Hurtig, "Feeling the beat where it counts: fostering multi-limb rhythm skills with the haptic drum kit," in *Proceedings of the fourth international conference on Tangible, embedded, and embodied interaction*, pp. 21–28, ACM, 2010. 1.2, 2.2

[4] E. Jaques-Dalcroze, *Eurhythmics, art, and education.* Arno Pr, 1930. 1.2

[5] B. H. Repp and Y.-H. Su, "Sensorimotor synchronization: a review of recent research (2006–2012)," *Psychonomic bulletin & review*, vol. 20, no. 3, pp. 403–452, 2013. 1.3.1, 1.3.3, 1.3.3

[6] S. A. Finney, "Ftap: A linux-based program for tapping and music experiments," *Behavior Research Methods, Instruments, & Computers*, vol. 33, no. 1, pp. 65–72, 2001. 1.3.2

[7] M. T. Elliott, A. E. Welchman, and A. M. Wing, "Mattap: a matlab toolbox for the control and analysis of movement synchronisation experiments," *Journal of neuroscience methods*, vol. 177, no. 1, pp. 250–257, 2009. 1.3.2

[8] B. G. Schultz and F. T. van Vugt, "Tap arduino: An arduino microcontroller for low-latency auditory feedback in sensorimotor synchronization experiments," *Behavior research methods*, vol. 48, no. 4, pp. 1591–1607, 2016. 1.3.2, B.2

[9] R. Polak, J. London, and N. Jacoby, "Both isochronous and non-isochronous metrical subdivision afford precise and stable ensemble entrainment: a corpus study of malian jembe drumming," *Frontiers in neuroscience*, vol. 10, p. 285, 2016. 1.3.3

[10] A. Białuńska, S. Dalla Bella, and P. Jaśkowski, "Increasing stimulus intensity

does not affect sensorimotor synchronization," *Psychological research*, vol. 75, no. 1, pp. 43–53, 2011. 1.3.3

[11] V. Krause, B. Pollok, and A. Schnitzler, "Perception in action: the impact of sensory information on sensorimotor synchronization in musicians and non-musicians," *Acta psychologica*, vol. 133, no. 1, pp. 28–37, 2010. 1.3.3

[12] S. W. Keele, R. A. Pokorny, D. M. Corcos, and R. Ivry, "Do perception and motor production share common timing mechanisms: A correlational analysis," *Acta psychologica*, vol. 60, no. 2-3, pp. 173–191, 1985. 1.3.4

[13] B. H. Repp, "Rate limits of sensorimotor synchronization," *Advances in Cognitive Psychology*, vol. 2, no. 2-3, pp. 163–181, 2006. 1, 2

[14] M. T. Elliott, A. Wing, and A. Welchman, "Multisensory cues improve sensorimotor synchronisation," *European Journal of Neuroscience*, vol. 31, no. 10, pp. 1828–1835, 2010. 2.1.1

[15] P. Ammirante, A. D. Patel, and F. A. Russo, "Synchronizing to auditory and tactile metronomes: a test of the auditory-motor enhancement hypothesis," *Psychonomic bulletin & review*, vol. 23, no. 6, pp. 1882–1890, 2016. 2.1.2

[16] P. Ignoto, I. Hattwick, and M. M. Wanderley, "Development of a vibrotactile metronome to assist in conducting contemporary classical music," in *International Conference on Applied Human Factors and Ergonomics*, pp. 248–258, Springer, 2017. 2.3, 4

[17] L. Gan, Y. Huang, L. Zhou, C. Qian, and X. Wu, "Synchronization to a bouncing ball with a realistic motion trajectory," *Scientific reports*, vol. 5, p. 11974, 2015. 2.4

[18] Peterson, "BodyBeat Sync," 2018. 2.4.1

[19] Soundbrenner, "Soundbrenner," 2018. 2.4.1

[20] A. M. Stark, M. E. Davies, and M. D. Plumbley, "Real-time beat-synchronous analysis of musical audio," in *Proceedings of the 12th Int. Conference on Digital Audio Effects, Como, Italy*, pp. 299–304, 2009. 5.1.3

[21] S. Holland, R. L. Wright, A. Wing, T. Crevoisier, O. Hödl, and M. Canelli, "A gait rehabilitation pilot study using tactile cueing following hemiparetic stroke," in *Proceedings of the 8th International Conference on Pervasive Computing Technologies for Healthcare*, pp. 402–405, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2014. 5.1.4

[22] S. Choi and K. J. Kuchenbecker, "Vibrotactile display: Perception, technology, and applications," *Proceedings of the IEEE*, vol. 101, no. 9, pp. 2093–2104, 2013. A.1, A.1.1, 5

[23] A. Fishbach, I. Nelken, and Y. Yeshurun, "Auditory edge detection: a neural

model for physiological and psychoacoustical responses to amplitude transients," *Journal of Neurophysiology*, vol. 85, no. 6, pp. 2303–2323, 2001. 4

[24] T. D. Parsons, "Neurobiology: auditory fidelity," *Nature*, vol. 444, no. 7122, p. 1013, 2006. 4

[25] Adafruit, "Mini Push-Pull Solenoid," 2018. A.3.1