SIRTAH: Singing Reinforcement Training Applying Haptics

- A Wearable Device That Uses Fundamental Frequency Tracking To Provide Vibrotactile Feedback & Help Vocalists Sing On Pitch -

Master Thesis Report

Michael Hedges

Aalborg University
Electronics and IT

Electronics and IT
Aalborg University
http://www.aau.dk

AALBORG UNIVERSITY
STUDENT REPORT

Title:
SIRTAH: Singing Reinforcement Training Applying Haptics

Theme:
Master Thesis

Project Period:
Spring Semester 2022

Participant(s):
Michael Hedges

Supervisor(s):
Stefania Serafin

Copies: 1

Page Numbers: 61

Date of Completion:
May 24, 2022

Abstract:

This study looks into whether a device that provides real-time vibrotactile feedback of the pitch accuracy of a singer is viable amongst many groups of people. The wearable device, named SIRTAH (Singing Reinforcement Training Applying Haptics) is designed using a YIN-based algorithm for fundamental frequency (f_0) or pitch estimation in cooperation with haptic feedback provided by a coin vibration motor. Two evaluations are conducted on different groups of novice singers (n = 23 and n = 6, respectively) in either a short-term or long-term trial with the goal of adjusting their singing until they've achieved near perfect pitch. Their evaluations are submitted through a System Usability Scale and their performance of accurately sung notes in different conditions are measured. The results indicate a passable level of usability, but suggests that adjustments and improvements could be made to achieve viability.

Elektronik og IT
Aalborg Universitet
http://www.aau.dk

AALBORG UNIVERSITET
STUDENTERRAPPORT

Titel:
SIRTAH: Sangforstærkningstræning i Anvendelse af Haptik

Tema:
Master Thesis

Projektperiode:
Forårssemestret 2022

Deltager(e):
Michael Hedges

Vejleder(e):
Stefania Serafin

Oplagstal: 1

Sidetal: 61

Afleveringsdato:
24. maj 2022

Abstract:

Denne undersøgelse undersøger, om en enhed, der giver real-time vibrotaktil feedback af tonehøjden for en sanger, er levedygtig blandt mange grupper af mennesker. Den bærbare enhed, kaldet SIRTAH (Singing Reinforcement Training Applying Haptics) er designet ved hjælp af en YIN-baseret algoritme til grundlæggende frekvens (f_0) eller tonehøjdeestimering i samarbejde med haptisk feedback leveret af en møntvibrationsmotor. To evalueringer udføres på forskellige grupper af begyndersangere (henholdsvis n = 23 og n = 6) i enten et kortsigtet eller langsigtet forsøg med det mål at justere deres sang, indtil de har opnået næsten perfekt tonehøjde. Deres evalueringer indsendes gennem en System Usability Scale, og deres præstation af nøjagtigt sunget noder under forskellige forhold måles. Resultaterne indikerer et acceptabelt niveau af brugervenlighed, men antyder, at der kan foretages justeringer og forbedringer for at opnå levedygtighed.

Contents

Preface

This study looks into whether a device that provides real-time vibrotactile feedback of the pitch accuracy of a singer is viable amongst many groups of people. The wearable device, SIRTAH (Singing Reinforcement Training Applying Haptics), is designed using a YIN-based algorithm for fundamental frequency (f_0) or pitch estimation in cooperation with haptics provided by a coin vibration motor. The signal generated by a singer has its pitch estimated, then compared against a list of equally tempered western music notation frequencies, of which the difference is output to the vibration motor to provide feedback. This study briefly explores the history of pitch estimation, with a focus on the YIN algorithm, and also looks at haptics and state-of-the-art technologies that surround these concepts. Development of the device will be discussed, including hardware, software and design. Two evaluations - short-term (20 minutes) and long-term (one week) - are conducted on different groups of novice singers ($n = 23$ and $n = 6$, respectively). Their evaluations are submitted through a System Usability Scale and their performance of accurately sung notes in different conditions measured. The results indicate a passable level of usability, but suggests that adjustments and improvements should be made to acquire greater viability. Discussions of the results, conditions that occurred during the process and future development conclude the study.

Aalborg University, May 24, 2022

Michael Carl Hedges
<mhedge20@student.aau.dk>

Chapter 1

Introduction

Singing is perhaps one of the most common instruments to be used worldwide thanks to the instrument itself being the human body. Nearly anyone has the capacity to sing or generate some sort of vocalization and this extends to many other species of animals as well. Its popularity tends to cause individuals to seek out improvement, whether it is through natural or technological means. When looking at natural methods of vocal training, there are some methods that are used to help improve one's voice to sing on key or on pitch. It would seem people mostly sing along to their favorite songs, while others seeking more professional assistance will be guided by a vocal coach or instructor. Vocal teachers use a variety of methods such as singing the notes themselves and having the student try to imitate what they have heard or playing notes on an instrument, such as a piano, and requesting the students attempt to vocalize a similar pitch. However, a lot of vocal pedagogy has begun focusing more on the anatomical functionality of the larynx and breathing techniques, suggesting that paying more attention to physiology is not only better for improvement in singing, but also healthier [38, 3].

Many aspiring singers turn to technological methods to either improve or modify their vocal capabilities. Over the past few decades, Auto-Tune - a digital signal processing (DSP) technique that shifts an audio signal to its nearest correct semitone and considered one of the most divisive elements to modern music - has critically and successfully allowed singers to sing on pitch through façade or creativity [8]. Others seeking a more honest means of singing on pitch from the comfort of their own home might turn to resources such as cellphone applications or other hardware devices that use pitch tracking algorithms and provide a visual reference for how near or far the sung voice is to a correct note. They may also use video games or karaoke machines, such as the ones described in section 2.1. These technological methods tend to utilize the auditory and visual sensory modalities through means of sensory motor coupling. There is limited, but growing research into another sensory modality for audio and music, that of touch and haptics.

When looking at what kind of roles visual, auditory and haptic stimuli play in music education and information retention, sensory dominance plays a big one. People using visual references, such as the tuner apps, video games or karaoke, may pay less attention to their voice and focus more on appeasing what they see in front of them and this is likely due to visual stimuli being more dominant than auditory stimuli [7]. In sensory coupling conditions, visual stimuli will continue to dominate auditory and haptic stimuli, whereas in auditory and haptic coupling conditions, neither tend to dominate the other [18]. This may be due to both audition (sound) and touch being sensitive to the very same kind of physical property, i.e., mechanical pressure in the form of oscillations.

Vibrotactile stimuli have proven to play an important part in the effectiveness of a musician's performance with a musical instrument. Though the physical materials of an instrument are vital to the timbre and musical output, the dynamic coupling between the biomechanical musician's interaction with the mechanical system of the instrument by feeling the quality of its resonance and kinematics, as opposed to simply hearing its output, allows them to be better suited to understand the utility and nuances of that instrument [35]. If the haptic elements of an instrument are removed, such as in the case of electronic instruments like synthesizers/digital piano and maybe even digital audio workstations (DAWs), some amount of loss occurs in the interconnections between musician and instrument. That being said, there are some discrepancies in the way people are able to utilize haptics. Some technologies, like the aforementioned digital piano, are looking into implementing haptics to generate natural piano-like sensations. The limitations to this comes in the sensitivity of our touch to the general frequency range. A study conducted by Federico Fontana et al. on the use of haptic simulation of grand and upright pianos on pianists suggested that the perception of vibrations was more prominent between notes A_0 (27.5 Hz) and A_4 (440 Hz), after which point the perception dropped significantly [10].

Another element of sensory modality that must be touched on is the lack of one. For individuals that are visually impaired, an auditory reference is useful, but not a visual one. For deaf and hard-of-hearing (DHOH) individuals, a visual reference is useful, but an auditory one is not. However, in both cases, touch-based stimuli could prove to be useful and research into these possibilities have been progressing since the mid 1900's. This brings about one of the other limitations of haptics: amount of exposure. In a study by Richard Miyamoto et al., two groups of hearing impaired children, ten subjects per group, were evaluated on whether speech perception skills were able to be enhanced through cochlear implants or a tactile device [30]. Both groups were tested at six month intervals and unsurprisingly, the results concluded that children with the cochlear implants performed much better than those using the tactile device. This case did also indicate that vibrotactile exposure over several years did show improvement in speech percep-

tion, suggesting that something is better than nothing. Be that as it may, the study narrowed the research to speech and even though speech is derived from the same source as singing, fundamental pitch perceptions are still diminished in hearing aid and cochlear implant users [26]. At this point, it must be asked: is it possible to improve vocal performance, or more specifically learning to sing near perfect pitch, by using a vibrotactile haptic feedback device and can it also be useful for DHOH individuals? Research in this study looks at what kind of role pitch tracking and haptics technologies could play in doing so.

This report will present some of the state-of-the-art music-based technologies that have integrated pitch estimation and haptics, and that have inspired this project. It will continue by elaborating on haptics and its physiological and technological aspects, followed by an explanation of fundamental frequency (or pitch) estimation; providing a brief history before delving into the type of algorithm used in this study. The SIRTAH (Singing Reinforcement Training Applying Haptics) device will be broken down into detail, exploring the hardware, software and design aspects that make up its functionality. The device will be evaluated in short and long-term circumstances by a group of participants. The participants will grade the usability of the device and measurements will be made on whether the device had any positive impact on their sung pitch accuracy. This report will conclude with a discussion of the development and evaluation process and what future implementations could be made to bring another dimension of usability to SIRTAH.

Chapter 2

State of the Art

Considering no other pitch detection-based vibrotactile feedback technologies were identified during the development of this study, the current state-of-the-art technologies that could be related will have to be discussed as being either based around pitch tracking or music-based haptics. Visual examples of the technologies discussed in sections 2.2 and 2.3 can be accessed in appendix C.

2.1 Singing Evaluators

When looking at technologies that employ some means of pitch tracking, the most abundant of them appear to stem from karaoke or singing performance-based entertainment. In video games, the concept of having a performer's singing evaluated through gamification popularized with games such as Karaoke Revolution®[1], Rock Band®[2] and Guitar Hero: World Tour®[3], where singers have to follow along to music with on-screen lyrics and notes in pitch and duration in order to receive a score. The popularity of these games caught the attention of the karaoke industry and more advanced and interactive systems were developed to draw more people out of their living rooms and into karaoke clubs and bars.

2.1.1 Almost Human: Karaoke Judgement

A project by Wei-Ho Tsai and Hsin-Chieh Lee, from the National Taipei University of Technology, looks at how to improve the performance evaluation system in modern karaoke devices so that it is comparable to human judgement [43]. They generated two databases; the first database took 20 Mandarin tracks extracted from a karaoke compact disc (CD) and the second was of 25 volunteers of three different

[1] `https://www.mobygames.com/game/ps2/karaoke-revolution`

[2] `http://www.harmonixmusic.com/games/rock-band/`

[3] `https://guitarhero.fandom.com/wiki/Guitar_Hero_World_Tour`

degrees of singing experience (professional, recreational and no experience). The evaluation system took into account three factors: pitch, volume and rhythm. In order to obtain pitch or fundamental frequency, they used Sub-Harmonic Summation (SHS) [36]. Spectral subtraction was needed to extract the vocal performances from the accompanying instrumentals taken from the CD source. The values provided from SHS were then converted into MIDI values, so the proximity to pitch from the volunteers could be compared. For volume, they used the estimated energy sequence from the vocal extraction. For rhythm, they took the vocal extraction and converted the waveform into a sequence of strength vectors and represented them by a hidden Markov model [37]. The performances were individually rated by four professional musicians who had their scores averaged to generate a reference score. The results showed that despite subjective differences, their system was similar to the professional musicians' evaluation, particularly in the classification of the level of singing experience.

2.1.2 Singing Quality: Good or Bad?

Inspired by the karaoke judgement project, Chitralekha Gupta et al. evaluated a group of singers against each other, singing the same song, rather than using a subjective panel of judges [14]. Instead of looking at the quality of voice - as compared to the original singer(s) of the song - note duration, accuracy, rhythm and location is evaluated, suggesting that the performances were not evaluated against a standard reference. They hypothesize that good singers will share characteristics of singing style and accuracy whereas poor singers should sing differently from each other. By utilizing a database of a cappella recordings, 50 male and 50 female vocal performances were subjugated to their system of measurements. Additionally, the *system* was graded against evaluations made by professional and semi-professional (in music comprehension) human judges. The performances were ranked from 1 to 100, where 1 is considered the highest rank and 100 being the lowest. These rankings were determined by the distribution of data points around a center, where condensed clusters indicated good singing quality and dispersed clusters indicated poor singing quality. The rankings suggested strong similarity between the implemented system and the subjective evaluations of the human judges.

2.2 Music Haptics

One of the more common applications of haptics in music is towards the use of pulse-based feedforward communication, usually tempo or guidance information. This type of haptics application has shown to elicit eventual entrainment in rhythm and coordination studies, such as the ones presented in the subsequent sections.

2.2.1 Haptic Bracelets

The Haptics Bracelets (figure C.2a) were designed by Bouwer, Holland et al. to provide coordination for percussive instruments, particularly a drum set, that require all four limbs. In an initial study, participants were asked to wear vibrotactile Velcro® bracelets around their wrists and ankels [5]. They were asked to play different sets of polyrhythms (deemed difficult for beginners and some intermediate players) on a MIDI drum set, which allowed for data collection. Results were generally positive about using haptics as a means of detecting rhythmic patterns, despite the hindrances of the bracelets being wired to a computer. As to whether performance was enhanced through the bracelets was yet to be determined. In the second study, a drum teacher would be wired to one set of bracelets and a student to another set [20]. The information generated by the teacher would be transfered to the student to mimic. The study looked at the effectiveness of rhythm education, whether assisted by audio, audio and haptics, or just haptics. In the same study, the teacher was removed to see if the same conditions applied showed any difference. The results reflected a preference in favor of using the haptics bracelets over the audio instruction.

2.2.2 Vibrotactile Metronome

A study by Marcello Giordano and Marcelo Wanderley looks into how well a vibrotactile metronome could be used to help musicians perform on tempo [13]. An armband with an actuator (figure C.2b), driven by an Arduino Mini Pro, sent out vibrational signals to the performer (four guitarists, individually) at 60 or 120 beats per minute (BPM). For comparative measures, the guitarists were also given an auditory metronome to follow along with. The participants were measured on their response to the stimuli or the amount of delay time between the occurrence of the auditory and vibrotactile signals and the picking/plucking of the guitar string. The auditory and vibrotactile metronomes were time synced using Max/MSP®[4] and the guitarists' performances were recorded. The conductors of the experiment discovered that there was a slower response time when using the vibrotactile metronome, but that the guitarists were able to pick up on the tempo as effectively as when listening to the auditory metronome.

2.3 Additional Similar Technologies

2.3.1 VEST

Headed by Scott Novich from Rice University in Houston, Texas, the Versatile Extra-Sensory Transducer, or VEST (figure C.3a), was developed as a wearable

[4] `https://cycling74.com/products/max-features`

haptics device that receives incoming sounds, analyzes its frequencies and sends the information to several actuators positioned throughout the vest [34]. VEST trains its wearer, accompanied by a cellphone application, on the vibrations they should expect from certain voiced words. Tested on DHOH individuals, the responses of the VEST from the participants were generally positive, some saying that it was preferred over their cochlear implant. VEST was further developed by a team headed by Dr. David Eagleman at the Bayer College of Medicine to make VEST more widely available and affordable. This project is inspirational towards what viability the device in this study could have towards DHOH individuals.

2.3.2 Breathing Reinforcement & Posture Control

In order to assist with breathing techniques associated with singing and vocal training, a haptics device was designed and tested by Yinmiao Li et al. [25]. The device consists of a system of straps that drapes over the shoulders and around the abdomen (figure C.3b). The abdomen strap is attached to a motorized gear system, which tightens and relaxes the area near the diaphragm to guide inhalation and exhalation. Additionally, a "spinal exoskeleton", consisting of three serial bus servos, manipulates the curvature of the back to further aid in the different breathing postures dependent on moments of inhalation and exhalation. The data sent to the microcontroller from the computer takes a MIDI file with manually labeled inhalation and exhalation duration and points, which is then interpreted by the device to execute the belt and spinal exoskeleton positions. Tested on participants with both breathing technique experience and lack thereof, the majority found validity and usefulness in the device.

Chapter 3

On Haptics

Haptics is the science of touch. It is a complex term attributed to both physiology and technology, but ultimately pertains to touch sensations. One of its most common uses today is in entertainment and communication technologies, such as video game controllers, to provide an extra dimension of playability, and cellphones, to indicate a received message, alarm or various other utilities. Haptics have played a quiet role in the way we navigate life, but various fields of science have started providing more of their attention towards touch-based technologies to enhance many people's way of life.

3.1 Touch Receptors

Mechanoreceptors that exist in the skin are linked to the nervous system and help with the perception and sensation of touch. In the field of haptics, especially in this study, two particular mechanoreceptors stand out: the Pacinian corpuscles and the Meissner's corpuscles (as seen in figure 3.1). These receptors operate congruently to signal to the brain different experiences, like texture, pressure and vibration [44].

Pacinian Corpuscles

The Pacinian corpuscles are deeper under the skin, in the subcutis (or subcutaneous) layer, and are larger receptors that respond well to vibrations and sudden pressure changes. Their peak sensitivity is around the 250 Hz frequency range; a frequency that will be used in this study (see section 5.2.1). These receptors can be found in both hairless and hairy locations of the skin.

Meissner's Corpuscles

Meissner's corpuscles exist closer to the surface of the skin, are much smaller and are more numerous. They are concentrated in hairless areas of the skin, such as

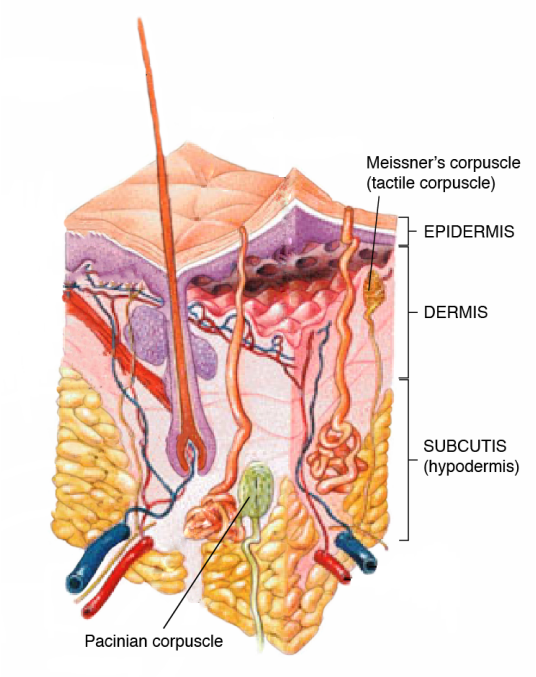


Figure 3.1: General location and appearance of the Pacinian and Meissner's corpuscles in the skin. Image was editied from its original source [31].

the palms of the hand and feet, the tongue, lips and genitals. Their peak frequency sensitivity is within the range of 10 and 50 Hz.

3.2 Perception Studies

In this study, the device that will be described in chapter 5 is a haptics-based feedback device that should be worn around the base of the neck. The location in which the singer will receive the vibration was chosen so that the adjacent microphone would be close enough to the signal source, since it operates as one unit, and to allow hands-free use of the device. Because more sensitive touch sensors are located in the palm of the hands, lips, etc., some research had to go into whether or not the neck area would be a reliable location to receive haptic signals. No experimentation regarding neck area vibrotactile reception was conducted in this study and assumptions were based on the following published studies. Images of the technologies described can be accessed in appendix C.1.

In an experiment conducted by Daniel S. Harvie et al., a wearable vibration-based device and four sensory tests were performed to measure the touch and pain acuity of the neck [17]. 22 participants were each subjected to four types of tests: two-point discrimination, point-to-point, graphesthesia and localization.

Two-point discrimination (TPD) utilized a caliper, placing two of its points on the neck, and subsequently adjusting the caliper until the participant was able to detect two distinct points. Point-to-point (PTP) required the participant to point (with a pen) to a position on the back of their neck where they perceived a sensation, which was generated by a Von Frey filament. The graphesthesia test had the participant verbalize the alphabetical letter they perceived being drawn on their neck with a pen (not specified whether the pen was capped or not). The localization test utilized a vibration device consisting of 12 vibrotactile nodes, which were localized to the back on the neck. The participants were then given a computer tablet with a picture of the back of their neck, covered in circles that indicate the points of the vibrotactile nodes. Whenever they perceived a vibration, they were to touch the circle on the tablet associated to the point on their neck where the vibration was perceived. The experiment was intended for clinical use in neck pain, but saw some errors that would indicate the test should be seen as a stepping stone. Three of the four tests showed similar results, with graphesthesia being the non-useful outlier and TPD showing the most promising results. The localization device showed near equal promise to TPD and PTP, which could indicate some validity in touch acuity of the SIRTAH project. However, this experiment looked at vibrotactile attention to the back of the neck, whereas the the vibration feedback provided by SIRTAH is intended to be localized to the collarbone area.

In a study by Rei Sakuragi et al., analysis of which areas of the body were able to best perceive vibrations was conducted [39]. Initial tests looked at four areas of the body with thin tissue-to-bone contact as candidates: the collarbone, the ribs, the scapula and the elbow. The collarbone proved to be the most efficient in perception of vibration for the purpose of their experiment. The next tests for perception of music through the whole body compared the collarbone, the head, the buttocks and palm of the hand. Alongside headphones to listen to the music, the vibrotactile devices were placed in said regions, with the collarbone device being fabricated via 3D printing. Participants were asked if they felt comfortable with the vibrations, if they felt vibrations through the whole body and whether the vibration changed the perception of music. Results indicated the collarbone and palm were close in comfort, collarbone conductance generated a greater whole body resonance for electronic and classical music as compared to the head and palm, and that the additional conductance to the collarbone and palms did alter/enhance the perception of experiencing music. Although the study focused on bone conductance as a source of music enhancement through haptics, the outcome, in regards to the SIRTAH study, indicates some level of haptics reception in the base of the neck in the chest and collarbone location.

Chapter 4

On Pitch Estimation

4.1 A Brief History

Pitch estimation is the process of finding the fundamental frequency (f_0) of a signal by means of an algorithm. It should be noted that fundamental frequency and pitch are not the same thing, but are used interchangeably to express a common concept. In a complex audio signal, one that is made up of multiple overtones (harmonics), the fundamental frequency is defined as the lowest resonant frequency of a periodic waveform [46]. Pitch is more of the perceptual properties of a frequency (or frequencies) that is able to be recognized, such as 'high' or 'low', within the audible frequency spectrum [24]. The reason fundamental frequency is so often referred to as pitch is due to musical connotations, where frequency is the physical property of pitch and pitch is the psychoacoustic property of frequency.

The method of pitch estimation used in this study is based on correlation, or more specifically, autocorrelation. Correlation refers to the process of comparing one or more seemingly random data sets to another, whereas autocorrelation compares the same data to itself. It has been speculated that the first published mathematical use and reference of correlation came from the physicist Auguste Bravais in 1844 [6]. Around the same time, Sir Francis Galton would develop his own method of correlation with anthropology, biology, heredity and psychology in mind [12, 42]. It would lead to the statistical concepts of regression and initiate the use of r as the correlation coefficient. He would later go on to devise the first concepts for autocorrelation. Galton's protégé, Karl Pearson, would expand on Galton and Bravais' concepts and develop the Pearson's correlation coefficient [23], as seen in equation 4.1.

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \overline{y})^2}} \tag{4.1}$$

where r_{xy} is the sample coefficient of paired data (x and y), n is the size of the

sample, x_i and y_i are points of individual samples within an index and $\overline{x}$ and $\overline{y}$ are equivalent to the sample mean ($\frac{1}{n}\sum_{i=1}^{n} x_i$). This equation suggests that data points equal to -1 or +1 lay exactly on a line in a graph and where values in the negative range have strong negative correlation, values in the positive range have strong positive correlation and zero value relates to no correlation. This would be used to support the concept of autocorrelation, which is elaborated on in section 4.2.1.

Norbert Wiener and Aleksandr Khinchin would expand on autocorrelation by producing a spectral decomposition when applying a power spectrum and formulating an analogous result for stationary stochastic processes [45, 21]; effectively generating the Wiener-Khinchin theorem, as expressed in equation 4.2.

$$\begin{aligned} R_{XX}(\tau) &= \frac{1}{2\pi}\int_{-\infty}^{\infty} S_{XX}(\omega)e^{j\omega\tau}d\omega \\ S_{XX}(\omega) &= \int_{-\infty}^{\infty} R_{XX}(\tau)e^{-j\omega\tau}d\tau \end{aligned} \tag{4.2}$$

This theorem suggests that the power spectrum (S_{XX}) can be obtained by taking the Fourier transform ($e^{-j\omega\tau}$) of the autocorrelation (R_{XX}) of a continuous signal and vice versa. This would catch the attention of Bruce P. Bogert, M. J. Healy and John W. Tukey, who would use this theorem to develop the concepts of 'cepstrum' and 'quefrency' (spectrum and frequency, respectively, with the first four letters inverted) [4]. Equation 4.3 expresses how cepstrum is, essentially, the spectrum of the logarithm of a spectrum.

$$C_p = |\mathcal{F}^{-1}\{log(|\mathcal{F}\{f(t)\}|^2)\}|^2 \tag{4.3}$$

where $\mathcal{F}$ is the Fourier transform and $f(t)$ is the signal over time. Bogert et al. used cepstrum to analyze periodicity in the frequency spectra of a signal or recording. Their analysis was implemented on the "echoes" or ripples generated by seismic activity. Although Bogert et al. concluded that cepstrum was not an effective means of analyzing seismic activity, their colleague Manfred Schroeder suggested that ceptrum analysis might be applicable towards voiced speech and vocal pitch determination. This caught the attention of A. Michael Noll, who rationalized that the process of cepstrum analysis made no mention of time length and that vocal sources are periodic and time dependent, in which case the spectral information can be analyzed through a window function; zero-valuing the outside of an interval or tapering the ends of a segment of a signal [33]. Repetition of the process allowed for a comparison of overlapping signals (one of which is delayed), identification of its maximum peaks and the points of delay between them, which would determine its f_0. Noll's research was intended to be implemented in vocoders and would consequently help influence alternative methods of pitch estimation.

4.2 YIN Method

The YIN alrogirthm was developed by Alain de Cheveigné and Hideki Kawahara in 2002 as a modification of the autocorrelation method of pitch estimation, in combination with a system designed to significantly reduce errors [9]. The system encompasses six essential steps: autocorrelation, a difference function (DF), a cumulative means normalized difference function (CMNDF), absolute threshold, parabolic interpolation and the best local estimate.

4.2.1 Autocorrelation

Equation 4.4 expresses the functionality of autocorrelation:

$$r_x(\tau) = \mathbb{E}[x(n)x(n-\tau)] \tag{4.4}$$

or alternatively expressed as:

$$= \sum_{n=\tau}^{N-1} x(n)x(n-\tau) \tag{4.5}$$

where $\mathbb{E}$ is the expected value of a signal sample $x(n)$ multiplied by a copy of the signal that has been offset by a number of samples $x(n-\tau)$. Tau (τ), sometimes referred to as the 'lag', is an unknown range of samples that progressively shifts a signal as well as the smallest amount of time a period of a signal repeats. An example of this process is shown in figure 4.1.

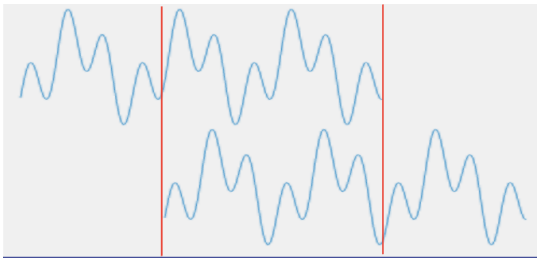


Figure 4.1: Autocorrelation: A signal (top) being compared with itself (bottom), which is shifted by a number of samples over time.

In signal processing, the algorithm discretizes the original signal from a continuous one into a finite one (essentially segmenting it), duplicates it and offsets the duplicate by a number of samples. The original signal and the duplicate are then compared against each other, one being shifted sample by sample through time, where it searches for the difference in peaks between the two signals. The differences produce a value which correlates the signals. For example, if the frequency is measured as time (T) equal to $\frac{1}{n}$ and the duplicate signal is shifted 440 samples

until its periodic peaks align with the original signal, then what is generated is $T = \frac{1}{400}$ or a 440 Hz frequency. Autocorrelation works well enough for periodic signals such as sine waves, but signals that are semi-periodic, like voices, require error reduction methods.

4.2.2 Difference Function

The difference function is written in terms of the autocorrelation function, as seen in equation 4.6, which searches for values of the lag (τ) that satisfies the function at zero.

$$DF(\tau) = \sum_{i=t}^{t+w} (x_i(n) - x_i(n+\tau))^2 \tag{4.6}$$

where t is time and w a window function (such as rectangular, gaussian, Hamming or Hann). Without a window, any deviations generated by the autocorrelation will be included in the sum. Squaring the difference function allows for prevention of cancellation caused by the positive and negative differences, as seen in figure 4.2. This is one step in reducing errors, but the algorithm requires more steps to find the best estimation of f_0.

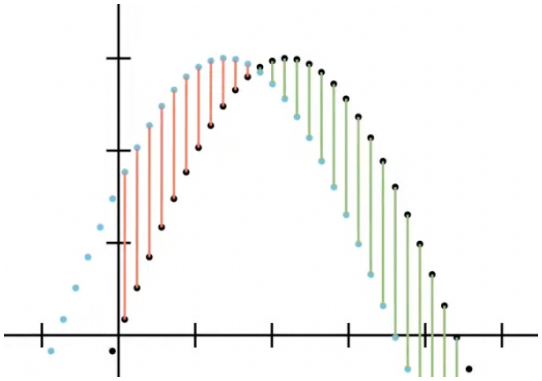


Figure 4.2: Representation of two of the same signal, one shifted in time, where positive and negative differences cancel out before squaring the difference function [40].

4.2.3 Cumulative Mean Normalized Difference Function

A cumulative mean normalized difference function, expressed in equation 4.7, replaces the initial difference function; starting with one instead of zero and effectively reducing the high value errors, cutting upper frequency limits and normalizing the function.

$$CMNDF(\tau) = \begin{cases} 1, & \text{if } \tau = 0 \\ \tau \cdot \frac{DF(\tau)}{\sum_{j=1}^{\tau} DF(j)}, & \text{if } \tau > 0 \end{cases} \tag{4.7}$$

The DF is normalized by dividing by the average over shorter τ values and defining the value at zero lag as 1. The CMNDF will only go below 1 when the DF is lower than the average of the prediction.

4.2.4 Absolute Threshold

The absolute threshold searches through the CMNDF for values that are over a threshold. It then picks the periods (or 'candidates') with the smallest positive value of τ or the global minimum in a set. This can be interpreted as the periodic areas tolerated within a semi-periodic signal, but depending on the sampling period, a gross error could be generated.

4.2.5 Parabolic Interpolation

Parabolic interpolation provides the set-up for the best local estimate by 'fitting' a local minimum with a parabola, where the y-axis points are used in a formant dip-selection process and the x-axis points serve as period estimates. It uses fractional shifts to the current τ estimate to calculate the first and second polynomial coefficients. This process is computationally more efficient than upsampling the signal because the peak period represents the same shape as the zero-lag peak and period dip. This process is not flawless and could come with complications, as will be described in section 4.3.

4.2.6 Best Local Estimate

The best local estimate portion of the algorithm acts as a sort of quality assurance measurement by using the current τ estimate and the first and second polynomial coefficients provided by the parabolic interpolation to determine which of the values is the most probable f_0 estimate. It does so by finding the minimum of an estimate of time and the largest expected period in a small interval and applies the process again, but with a reduced and restricted search range.

4.3 Project Validity

The YIN method would go on to be an industry standard for many applications and is still used today, due to its very low error rate and computational efficiencies. However, this is not to say that it is the best method for pitch estimation out there. The algorithm is still prone to errors, such as natural signals like speech or vibrato

singing being aperiodic or semi-periodic and when half cycle peaks are near in amplitude to cycle peaks, making it difficult to discern the frequency.

At the time of this writing, the algorithm is about 20 years old and since then, there have been vast improvements and innovations made to f_0 estimation algorithms. Probabilistic YIN (pYIN) [28] improves on YIN by implementing multiple possibility thresholds (instead of a singular absolute threshold) and applies a modified hidden Markov model to separate the pitch 'candidates' into bins along a finite frequency spectrum. This concept could have inspired one of the more modern f_0 estimation techniques of Convolutional Representation for Pitch Estimation (CREPE) [22]. This method uses machine/deep learning techniques to filter a signal through six convolutional layers that output to 360 nodes corresponding to a specific pitch value, from which the f_0 could be calculated. It was compared to other f_0 algorithms, such as pYIN, and produced a 99.9% accuracy rate.

CREPE was initially going to be used as the pitch estimation algorithm for this study. However, due to mathematical library limitations presented with the programming language used (see chapter 8 for more details) as well as necessary TensorFlow[1] model conversions to TensorFlow Lite[2] or TinyML[3], potential microcontroller memory limitations and computation costs, the YIN algorithm would provide the necessary means for a prototype evaluation. YIN still generates a high level of accuracy, especially in a project that does not require f_0 estimates to be as precise as possible.

[1] `https://www.tensorflow.org/`
[2] `https://www.tensorflow.org/lite`
[3] `https://www.tinyml.org/`

Chapter 5

SIRTAH

Figure 5.1: The final prototype of the SIRTAH device.

The SIRTAH device operates by a pulse-density modulation (PDM) micro-electromechanical system (MEMS) microphone receiving a signal (in this case, a singing voice), which is then delivered to a microcontroller. The microcontroller is programmed to interpret the f_0 of the received signal by processing it through a YIN pitch tracking algorithm. It then compares the f_0 to a list of musical note frequencies and generates a difference value. The value is used to send a pulse-width modulation (PWM) signal to a coin vibration motor to provide feedback to the singer. This will be further elaborated on in the subsequent sections.

5.1 Hardware

In this section, the hardware components of the SIRTAH device will be discussed in detail. It will describe the operational functionalities of each component as well as the parts that were 3D printed.

5.1.1 SIRTAH System

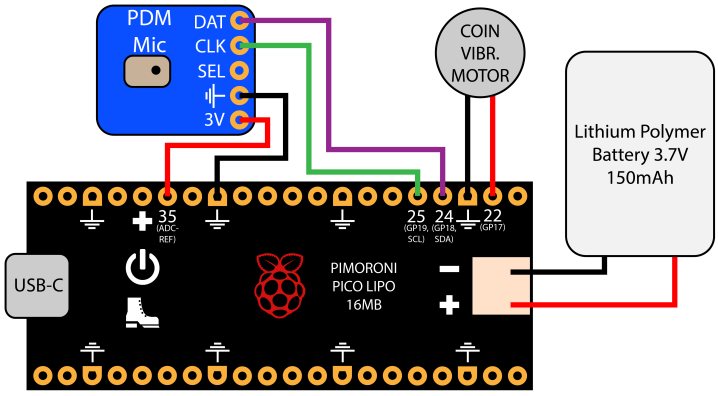


Figure 5.2: The basic schematics of the device. A Pimoroni Pico Lipo® microcontroller communicating with a PDM MEMS microphone, coin vibration motor and powered by a LiPo battery.

The microcontroller will run all of the pitch estimation and tracking processes from the code that can be found in appendix A. The PDM MEMS microphone is connected to pin 35 for reference voltage (3.3 volts), ground, pin 25 (GP19) for the serial clock (SCL) and pin 24 (GP18) for the serial data (SDA). The coin vibration motor is connected to ground and pin 22 (GP17), where the pin is programmed to send out a PWM signal. The lithium polymer (LiPo) battery connects to the molex connector faceted to the microcontroller.

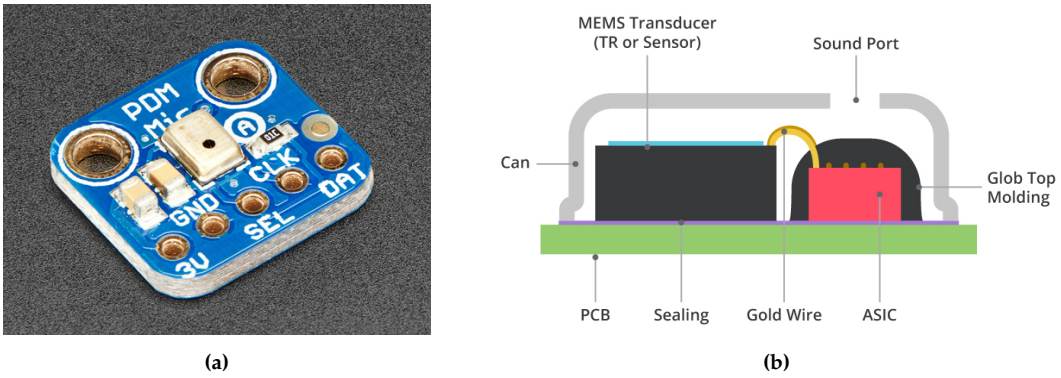
5.1.2 Microcontroller

The microcontroller chosen for this device is the Pimoroni Pico Lipo®[1] (depicted in figure 5.2), which is a versatile adaptation of the Raspberry Pi Pico®[2], as it uses the same RP2040 chip and has the same pinout. How it differs and why it was chosen over the latter is due to its inclusion of a USB-C data transfer and charging port (instead of micro-USB), 16 megabytes of flash memory, a 3.7v LiPo battery connection with recharging capabilities and a power on/off button. It has

[1] `https://shop.pimoroni.com/products/pimoroni-pico-lipo?variant=39335427080275`
[2] `https://www.raspberrypi.com/products/raspberry-pi-pico/`

a dual-core ARM Cortex M0+ processor and can run the programming language CircuitPython[3] (see section 5.2).

5.1.3 Microphone



(a) **(b)**

Figure 5.3: (a) The Adafruit PDM MEMS PCB® [1]. **(b)** A graphic representation of the components within the microphone [2].

The microphone used in this device is a MEMS microphone which uses PDM to send digital information of the incoming signal; a different method from the analog and I^2S methods of other microphones. Similarly to I^2S, it uses clock and data as sends to the input device. PDM is also similar to PWM in that it generates data like a square wave, which is synced to the clock. The data logic output is either 0 or 1, generating a density factor that can be averaged, which results in 'analog' values.

A MEMS microphone functions by having a freely moving diaphragm suspended above a backplate, which is fixed to a silicon printed circuit board (PCB). Sound pressure enters through the sound port (small hole) and causes the diaphragm to oscillate depending on the sound wave's amplitude. The variable distance between the diaphragm and backplate varies the capacitance. Then, a semiconductor converts the variable capacitance into an electrical signal, which outputs through the appropriate PCB pins. This microphone was chosen for its compactness and ability to turn analog signals into digital signals.

5.1.4 Vibration Motor

The type of haptic motor that is used in this device is an eccentric rotating mass (ERM) vibration motor, and more specifically, what is referred to as a 'coin' or 'pancake' vibration motor. Most other ERMs are cylindrical, whereas coin vibration motors are semi-flat, shaftless, circular plates, as the name might imply. The

[3] `https://circuitpython.org/`

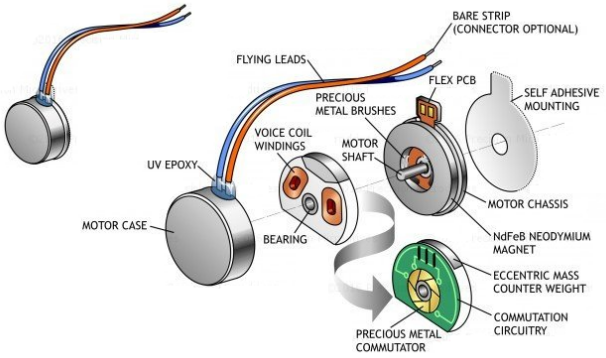


Figure 5.4: A graphic representation of a coin vibration motor and its functioning components [29].

vibration is produced through a mass offset, where one side of the commutator has a reduction in material and the other side uses a counter weight. It has two 'voice coils', which generate a magnetic field when provided power from the commutation PCB and rotates when reacting to the stationary magnet on the motor chassis. The coin vibration motor is ideal for small, wearable devices such as smart watches and smartphones.

5.1.5 Additional Parts

Battery

A lithium polymer battery is used for its recharging capabilities. 3.7 volts is all that is necessary to power the microcontroller and additional components. The one currently being used provides 150mAh (milliamp hours), which has proven to be sufficient battery life during testing of the device. Including a rechargeable battery allows for wireless portability and battery waste reduction.

Gooseneck

A 'gooseneck' (also known as a flex arm) is a flexible coiled metal hose or tube. It is commonly seen in use with lamps, microphone stands, faucets and various other fixtures. With SIRTAH, it is used to house and protect the wires that connect the microcontroller to the microphone and vibration motor. Its secondary function is to allow the wearer to reposition the microphone and vibration motor to where they deem it most comfortable.

It functions by having a round, helix shaped wire spiraled together with another wire with triangular cross-sections. When bent, it retains its shape due to friction between the two wires. The distance between the wires is less on the inner curve and greater on the outer curve, as seen in figure 5.5c.

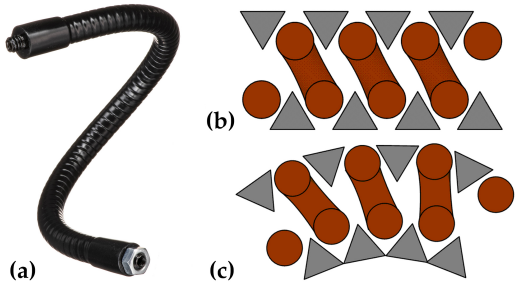


Figure 5.5: (a) A representation of the 'gooseneck'[15], **(b)** The coils straight, **(c)** The coils bent [47].

5.2 Software

The programming for SIRTAH has mostly been written in CircuitPython, a variant of Python developed for microcontrollers. The YIN algorithm [11] was written in C as a module - as was the majority of the boot loader file contents - which was wrapped in the CircuitPython build so that it could be read by the microcontroller. An explanation for the decision to use two programming languages can be found in section 8.1. The program operates from three main files: code.py, audioprocessing.py and noteprocessing.py. A more thorough explanation of the code can be accessed in appendix A.

5.2.1 Primary Operation

The generic file-naming convention of *code.py* is deliberate and mandatory; the microcontroller specifically searches for the *code.py* file to run its primary operation. This file intializes the PDM MEMS microphone, activates the pitch tracker and outputs the PWM signal to the vibration motor. It relies heavily on *noteprocessing.py* (section 5.2.3) to obtain difference values of the input frequencies that are compared against pre-defined musical note frequencies. These difference values are used to determine the vibration intensity by scaling the duty cycle of the PWM. The frequency of the vibration is maintained at 250 Hz to coincide with the peak sensativity of the Pascinian mechanoreceptors, as described in section 3.1. The program continually loops so that the feedback can operate in real-time. A more thorough explanation of the code can be accessed in appendix A.0.3.

5.2.2 Audio Processing

The main functions of *audioprocessing.py* are to generate a moving average, average the levels and notice when there is an input signal while powered on. Processing of the input signal begins when the microphone notices some level of activity based

on loudness. The microphone captures the input signal rapidly, so it requires a normalization of the root mean square (rms) in order to generate a more accurate reading of the signal.

The moving average takes several values generated by the YIN predictions of the input signal and average them out to create a smoother value. For example, if the YIN algorithm produces a series of values such as [441.1, 440, 439.7, 440.8, 441.5, 439.9, 440.4...], it will average the output value to something closer to 440. This is useful in creating a less erratic output. A more thorough explanation of the code can be accessed in appendix A.0.5.

5.2.3 Pitch & Note Processing

The function of *noteprocessing.py* is to take the values processed by *audioprocessing.py* and compare them against a list of equally tempered western music notation frequency values, which have been reduced to the singing range. In western music, singing ranges have been recognized as being between notes E_6 (low bass range) or 82.41 Hz and F_6 (high soprano range) or 1318.51 Hz [27]. Therefore, if the input frequency of 187.3 Hz is recorded, then it looks through an index where its nearest value of 185 Hz ($F\sharp_3$ / $G\flat_3$) is recognized. It determines the difference between these two values is 2.3 Hz and indexes it to be used to scale between the minimum and maximum vibration duty cycle values.

There are factors that play into the amount of vibration that is output. An array of frequency interval thresholds are established to represent the midpoint values between two notes. For example, the difference between frequency 185 Hz and 196 Hz is 11 Hz, which is the range between the two notes. To find the midpoint, which establishes the maximum amount of vibration, the range value is divided in half, making the example 5.5 Hz (+185, -196), or more specifically 190.5 Hz. A graphic representation of this can be seen in figure 5.6.

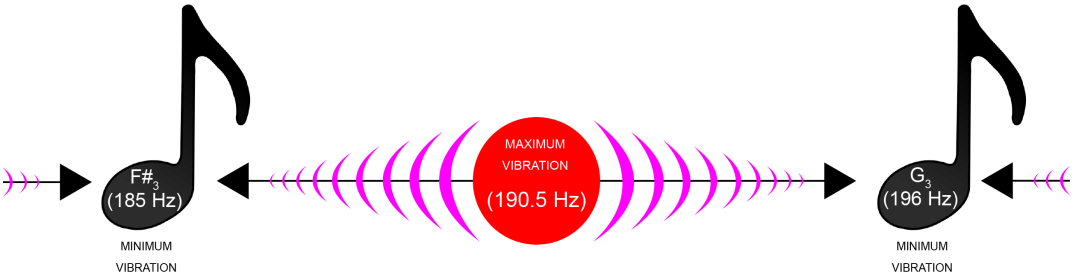


Figure 5.6: The midpoint between frequencies generates the strongest vibration. The vibration weakens the closer the input signal is to a note frequency value.

The minimum vibration amount is set to 20000 and the maximum set to 65535. The maximum vibration value represents $2^{16} - 1$, or 16 bit unsigned short inte-

ger used for the PWM output to the vibration motor (range = 0 to 65535). The minimum vibration value was chosen for its strength (or lack thereof) of vibration that could be sensed through the 3D printed material. Beginning the range at zero could have provided inaccurate feedback, since the intensity might not be noticed until nearly half of the way through the range.

Another factor is how the vibration is interpolated or scaled. For smoothness of the vibration intensity, a logistic function scaler is used [16, 19], providing a non-linear effect on the PWM output value. Equation 5.1 represents the logistic function used:

$$\sigma(x) = \begin{cases} \dfrac{1}{1+e^{-c(x-d)}} - \dfrac{1}{1+e^{cd}} & \text{if } x \geq 0 \\ -\dfrac{1}{1+e^{-c(-x-d)}} + \dfrac{1}{1+e^{cd}} & \text{otherwise} \end{cases} \tag{5.1}$$

where plateaus $c = 10$ and curve sharpness $d = 0.6$. Figure 5.7 shows a graphical representation of this equation.

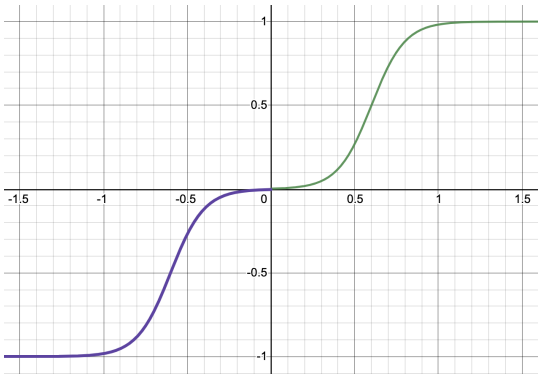


Figure 5.7: The shape of logistic interpolation used to control the smoothness and amount of vibration for feedback.

Finally, the relationship between input frequency and predetermined note frequency is indexed and the logistic interpolation applied to generate a value between 20000 and 65535 (the minimum and maximum vibration values, respectively). This value is obtained by *cody.py* to determine the PWM duty cycle, or vibration output, to be used in continual feedback for the singer. A more thorough explanation of the code can be accessed in appendix A.0.4.

5.3 Design

The hardware described in section 5.1 will need to be contained in an enclosure for protection and wearability. The following designs were developed using a free-

to-use online 3D modeling program called Tinkercad[4], which is straightforward and applicable towards 3D printing. All of the separate pieces designed for the SIRTAH device are shown in figure 5.8.

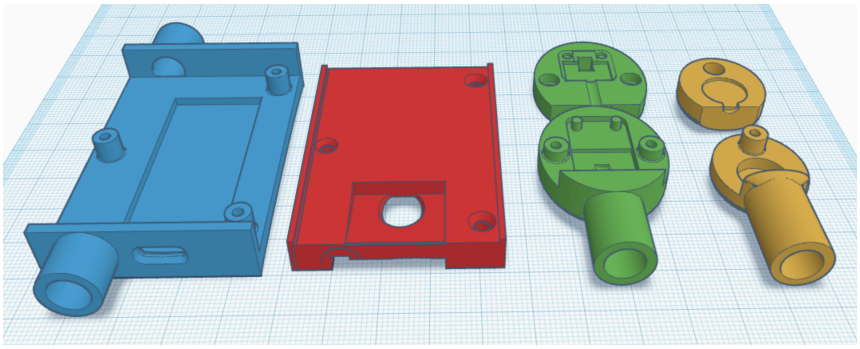
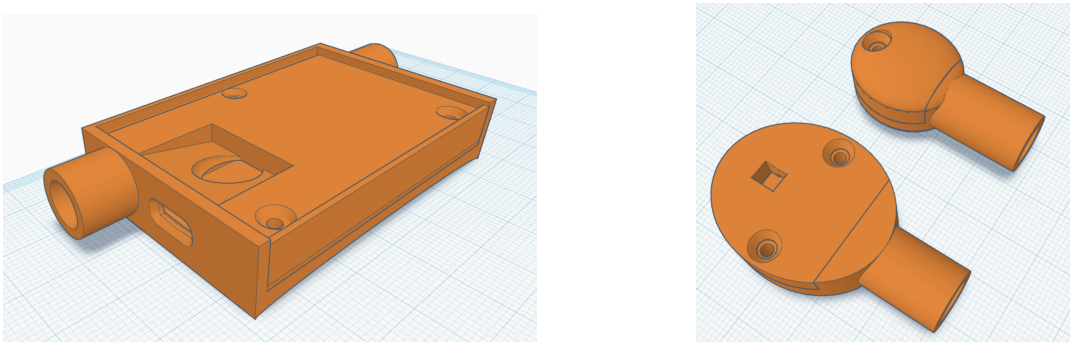


Figure 5.8: All of the separate pieces designed in Tinkercad for SIRTAH.

The main container (figure 5.9a) houses the microcontroller and the LiPo battery. It has a port for USB-C connection, holes for 3mm x 10mm screws, a top hole that provides access to the power button of the microcontroller and ports for the goosenecks, which the wires from the microcontroller will be led through to communicate with the PDM MEMS mic and coin vibration motor. The bottom of the primary enclosure is curved so that it can be placed comfortably on the back of the neck. The top of the enclosure is removable so that the microcontroller can be accessed for convenience of assembly.



(a) **(b)**

Figure 5.9: **(a)** The container for the microcontroller and battery. **(b)** The containers for the PDM MEMS microphone (bottom left) and coin vibration motor (top right).

The additional containers (figure 5.9b) were designed for the PDM MEMS microphone and the coin vibration motor. Like the main container, there are ports for the gooseneck, holes for screws and a removable top piece. The PDM MEMS

[4] `https://www.tinkercad.com/`

microphone container has a hole on the top so that sound is able to reach the microphone diaphragm. Additionally, small pegs protrude from the base so the PDM MEMS microphone can be held solidly in place.

5.4 Assembly & Duplication

The parts described in section 5.3 were printed using an Ultimaker 3®[5] 3D printer and printed with PLA[6] filament, a degradable material made from plant starches and sugars. The goosenecks were fitted into the parts container slots and adhered using epoxy resin. Wires run through the goosenecks, connecting the PDM MEMS microphone and coin vibration motor to the microcontroller. The connection points between the parts were further sealed using heat shrink tubing. Screws fastened the top parts of the 3D printed pieces to their respective component container parts. In total, four SIRTAH devices were developed with consideration that three would be used simultaneously during a long-term evaluation (see section 6.3) and the fourth would provide as a replacement if any of the other three were to malfunction. However, none of the devices were damaged or malfunctioned during the evaluation process.

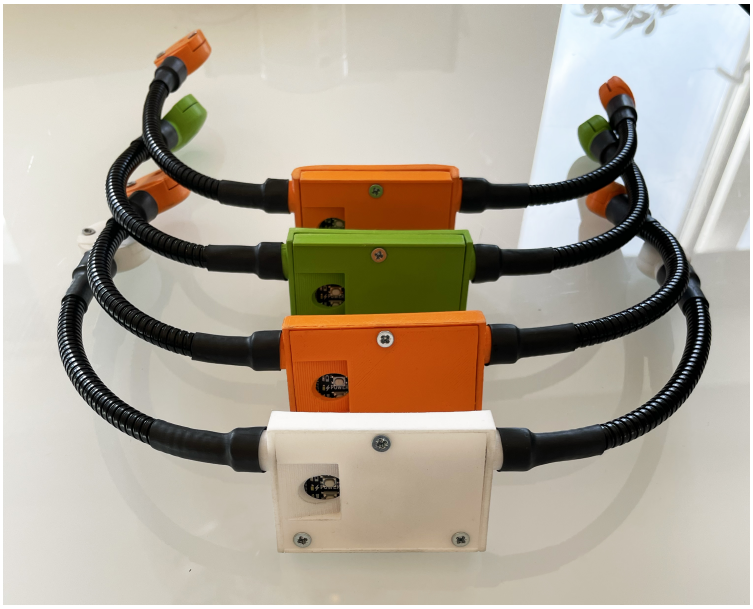


Figure 5.10: Four prototype SIRTAH devices were developed. Each one assembled using exactly the same components, but with different color enclosures.

[5] https://ultimaker.com/3d-printers/ultimaker-3
[6] https://stampomatica.com/is-pla-filament-biodegradable/

Chapter 6

Evaluation of SIRTAH

6.1 Operation

The functionality of SIRTAH is to detect pitch from the singer's voice, provide vibrotactile feedback and influence the singer to modify their singing depending on the feedback.

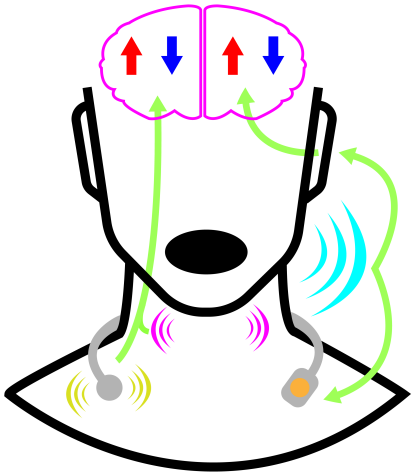


Figure 6.1: Intended sensory flow of the device and singer. Voice is heard by ear and device microphone, device interprets signal and outputs vibration feedback felt at the base of the neck, brain interprets vibration from device and larynx.

As the individual sings, their voice will be picked up by both the PDM MEMS microphone and in most cases, their ears. The signal obtained by the microphone is sent as data to the microcontroller - positioned on the back of the neck - and is evaluated through the YIN pitch detection algorithm and outputs a PWM value to the coin vibration motor, which vibrates in variable intensity based on the proximity

the f_0 of the voice is to a music note frequency. The vibration feedback is sensed through the skin (and sometimes heard), which sends a signal to the brain suggesting whether or not the singer needs to adjust their singing. This reinforcement should also help the singer to identify how their own voice feels in their larynx and how it sounds to their ear. Eventually, the singer might be able to sing without SIRTAH by recognizing how their voice sounds and feels when it is singing the [near] accurate pitch.

The effectiveness of the aforementioned process was evaluated in two separate conditions: either a short-term evaluation (about 20 minutes) or a long-term evaluation (about one week). As speculated in the introduction, haptic feedback may require some time to adjust to in order to become more acquainted with the type of information the brain is receiving. The short-term evaluation will utilize a large group of participants and provide them less time with the device in order to gauge immediate validity, whereas the long-term evaluation will give a small group of participants more time in order to gauge progressive validity. It is possible that neither times allotted may be sufficient enough for individuals to produce gains in pitch accuracy.

Both short-term and long-term evaluations were recorded and documented using an Audio-Technica AT2035®[1] cardioid condenser microphone with a pop-screen and a Focusrite Scarlett 2i4®[2] audio interface connected to a laptop running the Logic Pro®[3] digital audio workstation at a 44100 sampling rate. Simultaneously, a real-time pitch tracking program provided a visual reference, which was screen-captured. The files for this program can be found in appendix A.0.1. The participation was voluntary and there was no reward for partaking.

6.2 Short-Term

6.2.1 Participants

The participants (n = 23) of the short-term evaluation were two different groups of vocal students, all located in Hitra, Norway. The first group were teacher-student participants, ages ranged from 9 to 15. The second group were in a choir and ranged from ages 55 to 75. All participants were instructed by the same vocal teacher. One participant, age 30, was a colleague of the vocal teacher. Participants identified as either male or female, with 18 participants (about 78.3%) being female and five (about 21.7%) being male. Each participant followed a similar procedure.

[1] `https://www.audio-technica.com/en-us/at2035`

[2] `https://focusrite.com/en/usb-audio-interface/scarlett/scarlett-2i4`

[3] `https://www.apple.com/logic-pro`

6.2.2 Procedure

The participants were brought into the room where they normally have singing lessons. The conductor of the experiment explained to the participants, in English, what the device is, how it works and what they will be doing for the experiment. If the participants had a difficult time understanding what was said to them in English, their singing teacher would translate what was said into Norwegian. The participants were asked to sing into a microphone while their performance is being recorded and pitch-tracked on a computer.

First, the participants were asked to sing several notes, of their choosing, unaccompanied by the device or reference, such as a piano. This would act as a within-subject design by providing a control/baseline for the experiment.

Next, the participants were given the device to wear while being asked to sing a note that was played on a piano. In the initial instruction, the participants were told that the device would vibrate to some varying degree depending on how far or close they were to a note and that vibration would stop if they were singing on pitch. When a piano note was played, their task was to sing a sustained note and adjust the pitch of their voice up or down until the vibration stopped. The notes played on the piano were arbitrary and matched to the participant's vocal range.

The participants were then asked to sing several sustained notes, of their choosing, unaccompanied by the piano, while continuing to use the device. Their task was the same as before in that they were to change the pitch of their voice until the vibration stopped. Finally, they were asked to fill out a System Usability Survey (SUS)[4] that used a ten-point Likert scale, regarding their thoughts on the device. After doing so, the conductor of the experiment would leave the room so the participants could use the remaining time for their usual singing lesson.

6.3 Long-Term

6.3.1 Participants

The participants (n = 6) of the long-term evaluation were residents of Copenhagen, Denmark and five of the six participants were students at Aalborg University, average age around 28.5 years old. The sixth participant was a hard-of-hearing individual around 52 years of age. All identified as either male or female, with four participants (about 66.7%) being female and two (about 33.3%) being male. The participants had varying experiences of singing. Each participant followed a similar procedure.

[4] https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html

6.3.2 Procedure

The participants were invited into a general workspace at the Copenhagen campus of Aalborg University. The conductor of the experiment explained, in English, to the participants what the device is, how it works and what they will be doing for the experiment. The participants were first asked to sing several sustained notes, unaccompanied and unassisted by anything, while being recorded. This would act as a within-subject design by providing a control/baseline for the experiment. Next, the participants were given the device to become acquainted with the stimuli before being recorded. When they were ready, the participants were again asked to sing several sustained notes, but while trying to pay attention to the vibration feedback and adjust their voice as they deemed appropriate.

The participants were given the device to use over the course of a week, being asked to use it while singing for 10 to 20 minutes a day. Some asked whether they had to sing several sustained notes, similar to how they did during the recording, but it was advised that how they decided to sing was ultimately up to them. However, they were instructed to pay attention to the feedback provided by the device and work to make vocal adjustments accordingly.

After a week, they were invited back to the same workspace and were recorded performing the same procedure as before; singing several sustained notes unaccompanied by the device, then accompanied by the device. Afterwards, they were asked to fill out the same SUS questionnaire the short-term participants received, but this time adjusted to a five-point Likert scale. Upon completion, the participants were open to express any thoughts they had about the device and their time using it. Their verbal feedback was documented in paraphrased transcript, which can be accessed in appendix B.

6.4 CoolHear Workshop

Soon after the short-term evaluations, the CoolHear Workshop - held at the Royal Danish Academy of Music in Copenhagen, Denmark - presented an opportunity to demonstrate the functionality and intention of SIRTAH, alongside many other music and haptic technologies, to music students and experts in the DHOH community. Many had expressed interested in the device and had an opportunity to test their singing skills while using it. At one point, the device was dropped, but continued to function as intended, which suggests good durability or wear-and-tear of the hardware material and wiring. There were several individuals that thoroughly engaged in discussion about the potential of the device and the general response was interpreted as positive.

Chapter 7

Results

7.1 Short-Term Results

The results from the short-term experiment can be looked at in a number of ways and with some factors that play a part in the results (see section 8.2.1 for more details on the sample limitations). The SUS that the participants were given proceeding the experiment uses an alternating system of positively and negatively worded evaluators; odd numbered questions are worded positively and even numbered questions are worded negatively. The SUS uses a five-point Likert scale ranging from strongly disagree (1) to strongly agree (5), but in this experiment, the Likert scale used ten points (explained in section 8.2.1).

7.1.1 Using an Average

One way the data is interpreted is through an average score. The average computes the central value (mean) for the sample's responses in order to provide an overall indicator for the group. This score is then compared against a threshold to determine whether the system is usable or not. If the results were to comply with SUS score interpretation in order to be usable, then the score is expected to meet or exceed a baseline of 68 out of 100[1]. Figure 7.1 displays the results of different participant groupings that were averaged based on changing negative values into positive values (for example, a score of three in negative questions would equal a score of eight if positive) in a 100 point system.

What can be extracted from these results is that the device slightly exceeded the average baseline score of 68, meeting the criteria for usability on average with all participants and given a C grade, or considered "good"[2]. However, figure 7.1

[1] https://userfocus.co.uk/articles/measuring-usability-with-the-SUS.html

[2] https://about.gitlab.com/handbook/engineering/ux/performance-indicators/system-usability-scale/

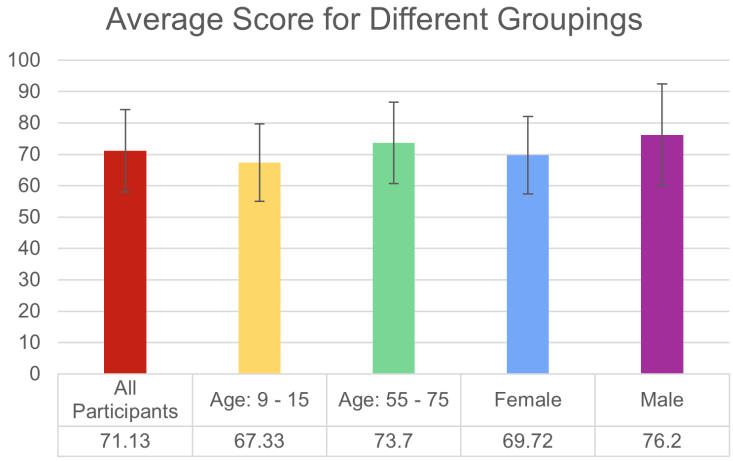


Figure 7.1: Averaged scores of different groups participating in the evaluation of the device.

weighs the response averages out of 100. By looking at the results as a percentile, the average score across *All Participants* is 71.13% and depending on which grading system is used, this is may generally be considered average.

Another statistical measure that should be considered is the standard deviation of the sample, which measures the spread of the responses received within the group and thus provides an indication of how far from the average most responses lie. The standard deviation of *All Participants* (s = 13.19) indicates that the majority (typically around two-thirds) of the participants provided an overall score between 57.94 and 84.32 (71.13, $\pm$ 13.19), with a minority (typically around one-third) of participants having scores that are lower or higher than those figures. The standard deviation is consistent across different age groups and among female participants, but it is higher (s = 16.18) among the male participants. This might be influenced by unbalanced distribution between the female and male groups, with a ratio of 18:5, respectively. Thus any single variation from the average response can have significant effect. That being said, between those identifying as female or male (none identified as non-binary, etc.), the device was generally more favored by males. When looking at the results between the 9-15 and 55-75 age groups, the adults rated the device more positively.

The standard error of the mean (SE) indicates how different the population mean can be from the sample mean of this group. It can be used to express a confidence interval that the population average is accurately reflected by the sample average. For example, with *All Participants* ($SE = 3.16$), a 95% confidence interval[3] can be calculated using the formula $1.96 \times SE$ (or $1.96 \times 3.16 = 6.19$). In this case, there is 95% confidence that the population average would be 71.13 ± 6.19.

[3] `https://academic.csuohio.edu/kneuendorfc53102/hand12.pdf`

7.1.2 Adjusting the Scale

Because this iteration of the SUS used a ten-point scale, rather than a five-point scale, the scores had to be adjusted using equation 7.1.

$$y = (B - A) \times \frac{(x - a)}{(b - a)} + A \tag{7.1}$$

where a and $A = 1$, $b = 10$, $B = 5$ and $x =$ the original value (1 through 10). This retains one as one and two through ten equally proportioned to two through five.

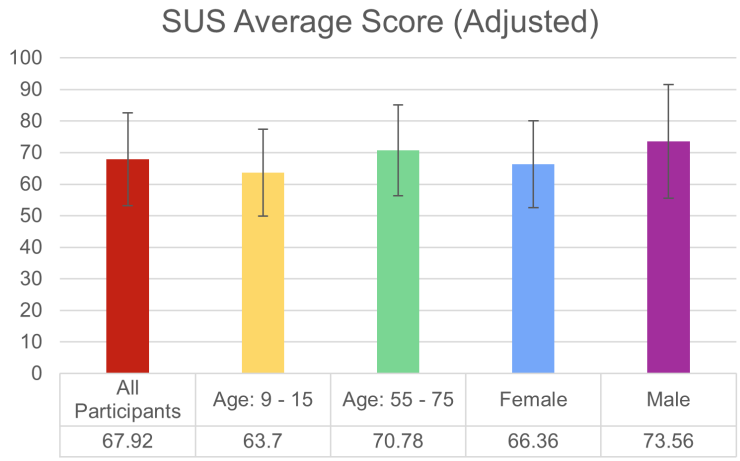
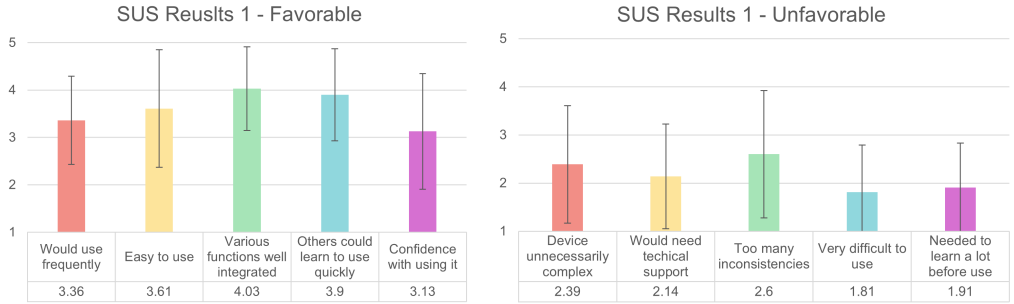


Figure 7.2: The SUS scores of participant groups after adjustments to scale by using equation 7.1.

The adjusted data represented in figure 7.2 shows that the usability of the device falls 0.08 points just below the SUS threshold average of 68, which could be considered indeterminate due to the adjustment. The *All Participants* standard deviation (s = 14.68) and standard error (SE = 3.51) reflect similarly to the non-adjusted average score. Another discrepancy with the results is the age gap between the sample, which does not accurately reflect the population, as is discussed in section 8.2.1.

7.1.3 Separating Questions (Short-Term)

When looking at the results of the individual SUS questions, it is easier to interpret by separating the questions into positively worded (favorable, figure 7.3a) and negatively worded (unfavorable, figure 7.3b) groups. The results were generally more favorable than unfavorable, with greater deviation in responses coming from the unfavorable questions results. Most participants thought the device had all of its functions and components well integrated and that operation and utility would be quick to figure out. However, the participants also felt that there were some inconsistencies and complexities in its usage.

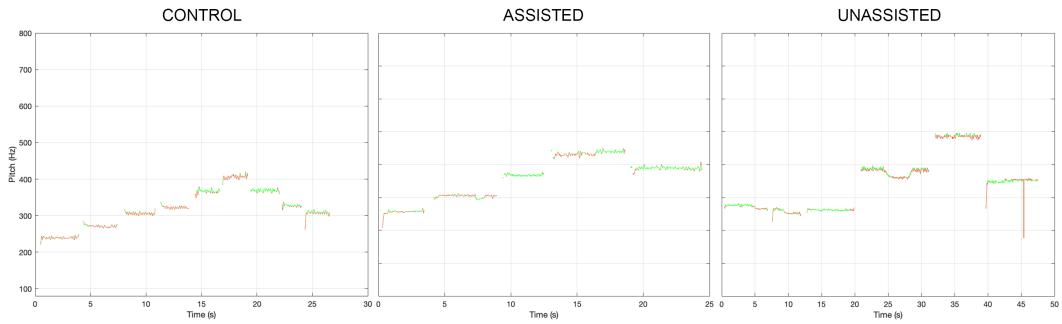


(a) **(b)**

Figure 7.3: (a) Results of the positively worded SUS questions. **(b)** Results of the negatively worded SUS questions. The higher the number in **(a)** and the lower the number in **(b)** the better the score.

7.1.4 Quantitative Measures (Short-Term)

During evaluation of the device, the participants' performances were recorded, as noted in section 6.2. The recordings were edited to remove conversation and other unnecessary noises between conditions and to produce smaller audio files. The audio files were processed through a MATLAB[4] script that estimates the fundamental frequency and colors areas that were on pitch in a different color, as shown in figure 7.4. The recordings are accessible through appendix D.0.4 and the MATLAB script that processed the recordings are accessible through appendix D.0.5.



(a) **(b)** **(c)**

Figure 7.4: The MATLAB plot example for participant 5A. The f_0 is drawn where red is the pitch and green is the pitch sung accurately. Graph **(a)** represents the recording without the device or any reference, graph **(b)** represents the device used and with a piano reference and graph **(c)** represents use of the device without a reference.

The quantitative measurement looked at how many notes were sung accurately of all sung notes during the three conditions: control measure (*Control*), with assistance from the device and piano reference notes (*Assisted*) or with the device, but without a piano reference (*Unassisted*). The difference between on-pitch and

[4] `https://se.mathworks.com/products/matlab.html`

off-pitch notes sang were given a percentage of accuracy. This was to determine whether the device had any immediate effect on how individuals might cognitively respond to it beyond their qualitative evaluations.

As mentioned in section 6.1, the gain from using such a device in the short-term may not be significant, but not implausible. The data collected from the short-term participants is treated as nominal, so analysis was carried out using a non-parametric Friedman's ANOVA[5]. Participants 1A through 3A did not have their data included due to not undergoing a control condition. Participant 10A was also not included due to an error in the recording. The amount of accurately sung notes while using the device was not significantly affected, $\chi^2(2) = 5.44$, ns. Hence, no follow-up analysis is necessary. This result supports that very little to no change is noticeable in the short-term use of the device.

7.2 Long-Term Results

The results from the long-term evaluation showed both similar and different results compared to the short-term evaluation.

7.2.1 Usability Average

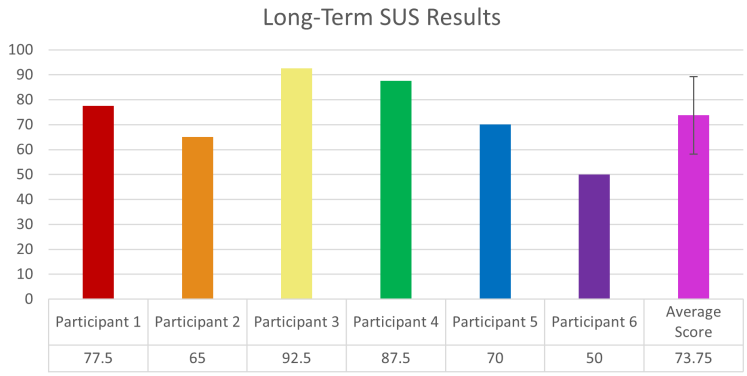


Figure 7.5: SUS results from the long-term participants, including the average from all results and the standard error of the sample.

Generally, the device was received somewhat more positively amongst the long-term participants than amongst the short-term participants. The long-term SUS average improved by 5.83 points, giving it a score of 73.75 and exceeding the threshold, suggesting the device has usability. The standard deviation (s = 15.55) and standard error of the mean (SE = 6.35) of the averaged score are greater, which might be due to the smaller sample size.

[5] `https://www.statology.org/friedman-test/`

7.2.2 Separating Questions (Long-Term)

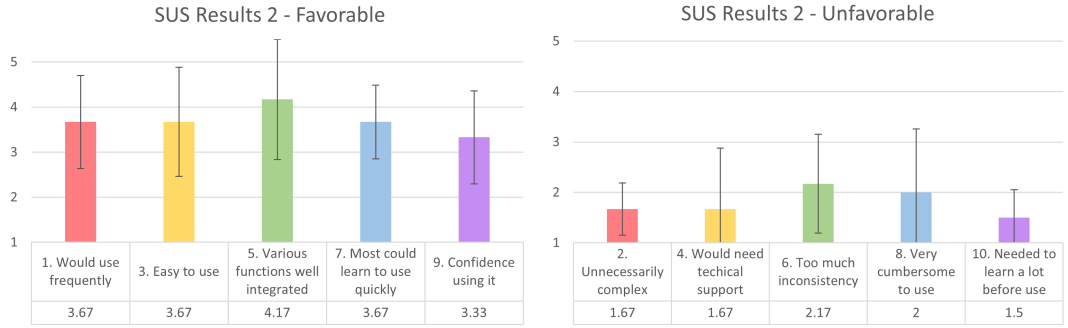


Figure 7.6: (a) Results of the positively worded SUS questions. **(b)** Results of the negatively worded SUS questions. The higher the number in **(a)** and the lower the number in **(b)** the better the score.

The SUS results for the long-term group have a very similar pattern as the short-term group. In terms of the favorable questions, both groups suggest the device's various functionalities were well integrated, but the participants had lower confidence using the device. Speculation as to why this may be can be found in section 8.2.2. The results of the favorable questions in the long-term evaluation showed an average of about 3.7 out of 5, or 74% favorable.

In terms of the unfavorable questions, both groups of participants agreed that the main issue was that there were inconsistencies with the operation of the device, although this was not a particularly pronounced concern overall. Neither group felt that they required much effort to learn about how to use the device, suggesting that operation was straightforward. The long-term group also suggest that the device is simple to use and that they would not require technical support to operate the device, as compared to the short-term group, who felt more strongly that it is unnecessarily complex and might require a technician. This might be due to the short-term group's discrepancy in age and/or the more technology-oriented education of the long-term group. The results of the unfavorable questions in the long-term evaluation showed an average of about 1.8 out of 5, or 36% unfavorable.

7.2.3 Quantitative Measures (Long-Term)

As with the short-term group, the long-term group had their performances recorded and analyzed through the same MATLAB script. The number of notes sung were counted and were contrasted against the number of those notes sung accurately or on-pitch and given a percentage. The percentages, as they pertain to sung pitch accuracy in both the *With* and *Without* the device conditions, before and after a week-long trial, are reflected in figure 7.7.

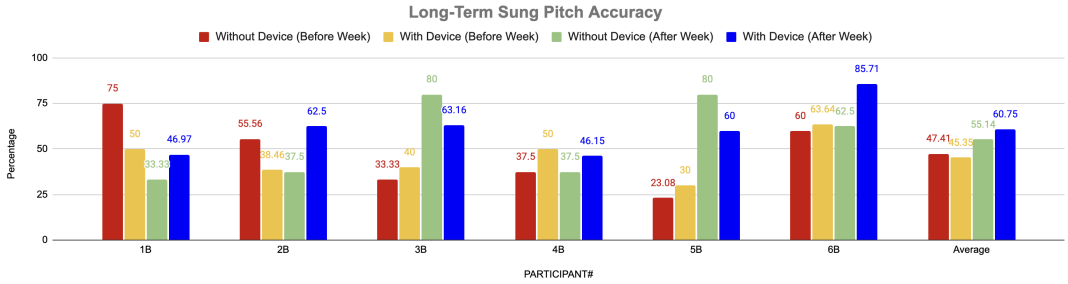


Figure 7.7: Percentages of individual long-term participants sung pitch accuracy results after a week of using SIRTAH, as well as the average between participants from analyzed recordings before and after their trial of the device.

An inference made by viewing the results is that there was some degree of improvement in sung pitch accuracy amongst the long-term participants. Further speculation as to why this may be is discussed in sections 8.2.2 and 8.3. Four particular participants stand out and should be discussed: participants 1B, 3B, 4B and 5B. Participant 1B chose to sing significantly more notes than their fellow participants: four notes in the *Without Device (Before Week)* recording, 40 notes with *With Device (Before Week)*, 21 notes with *Without Device (After Week)*, and 66 notes with *With Device (After Week)*. This, as well as a reported illness during recording, could have skewed the results of their accuracy. Participant 4B is also worth noting in that there was practically no change in their *Before Week* and *After Week*. As to why this is may be reflected in the transcription of their post-evaluation interview, which is accessible in appendix B.0.4.

Participants 3B and 5B stand out due to their *After Week* scores being higher than their *Before Week* scores. Participant 3B exhibited some degree of excitement before and after the trial, which is reflected in both their SUS score and their post-evaluation interview, which can be accessed in appendix B.0.3. It is plausible that this played a role in their willingness to use the device and may be reflected in their pitch accuracy results. Although participant 5B, the individual that is hard-of-hearing, provided one of the lower SUS scores, their pitch accuracy noticeably increased. It is possible that even though they may not have favored use of the device (reflected in their post-evaluation interview accessible in appendix B.0.5), they may have unknowingly benefited from it; however, this is purely speculative.

Chapter 8

Discussion

8.1 Complications

As with any prototyping project, some complications were likely to arise. In early stages of development, the original intention was to run the project entirely on Python. CREPE, the more accurate, robust and modern method of pitch estimation, was written in Python and relied on Python's NumPy library for mathematical operations. With CircuitPython, only a diminished version of NumPy called ulab is accessible and where some functions of NumPy are available in ulab, not all of them are. This caused complications where missing functions in ulab would need to be rewritten. The consequence of this is what led the project to utilizing a YIN-based algorithm, written in C, instead. Circumstantially, incorporating the YIN-based algorithm also permitted more storage on the microcontroller and may have allowed for faster processing, due to the computational load being lighter.

In terms of hardware, several occasions of 3D printing were needed. This was due to a number of factors, such as pieces breaking and needing to be redesigned, printing errors and interruptions where filament would not adhere or run astray, power to the printer would turn off due to some electrical outages on campus and measurement errors and oversights made during the design process. 3D printing is also a lengthy process, where the total print time for all parts was roughly 8 hours for just one device. Ultimately, 3D printing is a great way to prototype but it is also time consuming and not greatly beneficial towards a consumable product.

8.2 Perceptions During Evaluation

8.2.1 Short-Term Review

As noted in section 6.2, during the short-term evaluation, the participants were vastly different in age, making up the lower and higher ends of the age bracket.

The older group generally expressed more interest, excitement and viability towards the device, which may be reflected in the results (see section 7.1). The younger group were very shy and not as responsive or expressive, though some expressed to their vocal teacher that they thought the device could be useful. Some of the younger students with more experienced vocal training would sometimes not acknowledge much viability of the device.

Speaking on some of the SUS results, particularly those that emphasize lower confidence with using the device and lower likelihood of using it frequently, the response could be due to either a lack of confidence or motivation in oneself with singing in general or having limited time (20 minutes) experiencing the nuances of the device. Because the participants were given a short amount of time to use and experience the device, many felt that they were not given enough time to become accustomed to its vibration-based feedback. Most of the older participants expressed that if they were to have more time with the device, they could gain greater use of its potential in helping them sing on pitch, as well as acquiring comfort in perceiving the vibratory feedback.

It should also be noted that a language barrier may have played a role in the results. The SUS questions were provided in English, Norwegian and Danish; four participants took the survey in English and 19 in Norwegian. The younger participants were given English instructions, which were often iterated in Norwegian, and the older participants were given only English instructions, due to the teacher needing to conduct the remaining choir members in a separate room. Some terminology was not easily translatable, particularly with the younger participants and some of the procedural instructions were not easily conveyed to/understood by some of the older participants.

In regards to the adjustment of the SUS Likert scale from ten points to five points, this was due to an oversight. When the short-term participants were given the survey, the scale between strongly disagree and strongly agree was unintentionally set to a ten point scale and this error was not realized until after all participants had evaluated the device. The scale was proportionally adjusted from ten points to five points by using equation 7.1 and this seemed necessary due to System Usability Surveys using a five-point scale and for people's perception of the scale. Five-point scales allow for a center or neutral point to base their perceptions on, whereas ten-point scales are even numbered and force the survey-taker to choose more positively or negatively [32].

8.2.2 Long-Term Review

Most of the participants in the long-term evaluation were students of Aalborg University, as was mentioned in section 6.3. There could be indication of bias because of this, however, four of the six participants that were students had been met the

day of the introduction to the device and no contact was made until arranging for the final evaluation. Participant 6B did have a more casual acquaintanceship to the conductor of the experiment, but their evaluation of the device was deemed professional, critical and non-biased, as was evident in the SUS score and feedback they provided (see section 7.2 and appendix B.0.6, respectively).

Some participants and non-participant evaluators mentioned how it could be useful to have the device vibrate at the same frequency of the nearest accurate pitch, rather than the same 250 Hz frequency. This was taken into consideration during the initial development of the device, but was ultimately avoided. In the introduction, it was mentioned that perception of vibrations in digital pianos was more prominent between 27.5 Hz and 440 Hz, with perception diminishing significantly after this range [10]. Considering the frequency range from low bass (82.41 Hz) to high soprano (1318.51 Hz), this would work well for lower frequencies, but at some point in the higher range of frequency, vibration is less likely to be perceived and may cause confusion or inaccuracies; potentially leading people to think they are singing on pitch, due to the perceived lack of vibration, when they may not be accurate at all.

As mentioned in the introduction, musicians have a tendency to rely on haptic information in order to understand the nuances and utility of their musical instruments [35]. During the long-term evaluation, one of the participants reinforced this concept by mentioning how it felt unusual to *not* use the device while singing, as they had become reliant on the feedback information it provided (see B.0.3). In another instance, a different participant made notice of how, after a few days of using the device, it felt correct in their larynx and sounded correct to their ear when they were able to sing on pitch (see B.0.1).

One of the more valuable responses in the long-term study was that of the hard-of-hearing individual. Because the device was designed with DHOH individuals wanting assistance with singing in mind, participant 5B happened to fit this description. Upon review, they mentioned that at times they found the device somewhat difficult to work with, mostly noting that the feel of the device was not comfortable during use. Much like the participants in the short-term evaluation, they also specified that it required some time getting used to the haptic feedback and that more time with the device would be desired. Ultimately, they did not see much viability in the device, but did suggest that after some modifications, it should be tested on other DHOH individuals more enthusiastic about developing their singing skills.

8.3 Additional Thought

A final observation is that the quantitative results should not be considered substantial enough to indicate that the device could have improved the participant's

ability to sing. It is also likely that merely practicing singing more frequently could have been consequential to the improvement of their sung pitch accuracy results. Further evaluations under stricter conditions, a larger, more diverse sample and potentially longer use of the device (for example a month or half a year) should be implemented to determine the viability of the device.

Chapter 9

Future Work

9.1 Immediate Modifications

Between the short and long-term evaluations and other instances of presentation to the public, SIRTAH was generally well received and intrigued those with an opportunity to learn about it and try it for themselves. Many have spoken positively about the concept and provided examples of what could enhance it. One of those suggestions is to provide inverse feedback, where a vibration indicates that the wearer is singing on pitch, rather than off pitch. This could alleviate potential stress or frustration brought on by a constant vibration indicating the singer is off pitch. One factor that should be considered with this modification is whether the wearer would be confused by the lack of response from the device.

An alternative to this could be to provide different types of vibrations indicating whether the singer should adjust their voice up or down, considering there is no current type of indication and it is left up to the wearer to decide. Additional components, like the Adafruit DRV2065L Haptic Motor Controller®[1] can be incorporated to provide different types of feedback pulses; an example could be short, 'click'-like vibrations indicating to adjust one's voice upward and longer, 'hum'-like vibrations suggesting to adjust downward. During the design phase, this component was being used but was later removed in favor of variable intensity PWM output to the vibration motor.

Another suggestion was to utilize a pentatonic scale, rather than the chromatic scale that is currently programmed to the device. The fewer notes could provide a greater gradient between correctly and incorrectly sung pitches. On a different note, with further development, SIRTAH could utilize the CREPE pitch estimation algorithm over the YIN method, as was initially planned. This could provide an even more accurate response and highlight the machine/deep learning direction the industry might take.

[1] `https://learn.adafruit.com/adafruit-drv2605-haptic-controller-breakout/overview`

9.2 Alternative Modifications

It is possible to reduce the design of SIRTAH to something more ergonomic. An idea presented by a non-participant is to condense the device to something like a pin that can be worn anywhere on one's clothing. This could be done with greater resources, such as smaller microphones, smaller vibration motors, professionally engineered PCBs with powerful enough processing chips and better encasing materials. So long as the microphone were able to receive a loud enough signal, the SIRTAH pin could be worn in more convenient locations, rather than around the base of the neck.

In a similar study to this one, Larry Solberg et al. used different f_0 estimation devices to determine if there were more cost effective technologies that could aid in clinical speech therapy [41]. One of the potentials that was realized during the development of SIRTAH was the utility the device could provide in speech therapy and articulation, particularly for DHOH individuals. It is possible to take similar design features and incorporate a system that identifies voiced (vowel) sounds and measures their accuracy. This could be done through machine/deep learning processes which identify correctly pronounced vowel sounds (and maybe even to a lesser degree, unvoiced or consonant sounds) and provide haptic feedback indicating either to adjust the vocalizations and intonations or that they are speaking within some degree of accuracy. It could even be developed for individuals learning to speak languages with unique vowel sounds, such as with the different Scandinavian languages. With more time, development and testing, such a device could make an impact on many people's speech.

Chapter 10

Conclusion

The SIRTAH device was designed to assist in improving individual's singing skills by providing vibrotactile feedback through pitch estimation. This study shared existing and related technologies, made a case for haptics and elaborated on fundamental frequency estimation algorithms. The hardware and software designs provided insight into how SIRTAH is able to function. Evaluations took place, with 23 participants in a short-term trial and six participants in a long-term trial. Their evaluations were measured both qualitatively and quantitatively; suggesting some level of usability and viability. Their System Usability scores and post-evaluation interviews indicate that the device has potential, but could use some work before it provides the desired effect. Their sung pitch accuracy measures indicate that over a longer frame of use, the device might improve their ability to sing on pitch. Amusingly, early f_0 algorithms that measured seismic vibrations influenced a device that now generates vibrations from f_0 estimates. Research and innovations in pitch estimation, haptics, music technologies, their interconnections and how they may improve the everyday lives of individuals remain on-going and ever-growing.

Bibliography

[1] Adafruit and lady ada. *Adafruit PDM Microphone Breakout*. https://learn.adafruit.com/adafruit-pdm-microphone-breakout/, 2022.

[2] Majeed Ahmad. *How does a MEMS microphone work?* https://www.planetanalog.com/how-does-mems-microphone-work/, 2021.

[3] Gracie Bennett. "The Science of Singing: A Voice Lesson from Anatomy and Physiology". In: (2017).

[4] Bruce P. Bogert, M. J. R. Healy, and John W. Tukey. "The quefrency alanysis of time series for echoes; Cepstrum, pseudo-autocovariance, cross-cepstrum and saphe cracking". In: *Time series analysis* (1963), pp. 209–243.

[5] Anders Bouwer, Simon Holland, and Mat Dalgleish. "The Haptic Bracelets: learning multi-limb rhythm skills from haptic stimuli while reading". In: *Music and human-computer interaction*. Springer, 2013, pp. 101–122.

[6] Auguste Bravais. "Analyse mathématique sur les probabilités des erreurs de situation d'un point". In: (1844), pp. 255–332.

[7] Francis B Colavita. "Human sensory dominance". In: *Perception & Psychophysics* 16.2 (1974), pp. 409–412.

[8] Zachary Crockett. *The Mathematical Genius of Auto-Tune*. https://priceonomics.com/the-inventor-of-auto-tune/, 2018.

[9] Alain De Cheveigné and Hideki Kawahara. "YIN, a fundamental frequency estimator for speech and music". In: *The Journal of the Acoustical Society of America* 111.4 (2002), pp. 1917–1930.

[10] F Federico et al. "Perception of interactive vibrotactile cues on the acoustic grand and upright piano". In: *Joint ICMC-SMC Conference*. SMC. 2014, pp. 948–953.

[11] Ashok Fernandez. *YIN Pitch Tracking*. https://github.com/ashokfernandez/Yin-Pitch-Tracking, 2014.

[12] Francis Galton. "I. Co-relations and their measurement, chiefly from anthropometric data". In: *Proceedings of the Royal Society of London* 45.273-279 (1889), pp. 135–145.

[13] Marcello Giordano and Marcelo M Wanderley. "Follow the tactile metronome: Vibrotactile stimulation for tempo synchronization in music performance". In: *Proceedings of the Sound and Music Computing Conference. Maynooth, Ireland*. Citeseer. 2015.

[14] Chitralekha Gupta, Haizhou Li, and Ye Wang. "Automatic leaderboard: Evaluation of singing quality without a standard reference". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 28 (2019), pp. 13–26.

[15] B & H. *Marshall Electronics Gooseneck Arm Mount*. `https://www.bhphotovideo.com/c/product/1356909-REG/marshall_electronics_cvm_13_10_flexible_gooseneck_arm.html`, 2021.

[16] Jun Han and Claudio Moraga. "The influence of the sigmoid function parameters on the speed of backpropagation learning". In: *International workshop on artificial neural networks*. Springer. 1995, pp. 195–201.

[17] Daniel S Harvie et al. "Tactile acuity testing at the neck: a comparison of methods". In: *Musculoskeletal Science and Practice* 32 (2017), pp. 23–30.

[18] David Hecht and Miriam Reiner. "Sensory dominance in combinations of audio, visual and haptic stimuli". In: *Experimental brain research* 193.2 (2009), pp. 307–314.

[19] Andre Hilsendeger et al. "Navigation in virtual reality with the wii balance board". In: *6th workshop on virtual and augmented reality*.

[20] Simon Holland, Anders Bouwer, and Oliver Hödl. "Haptics for the Development of Fundamental Rhythm Skills, Including Multi-limb Coordination". In: *Music Haptics*. Springer, Cham, 2018, pp. 215–237.

[21] Alexander Khintchin. "Korrelationstheorie der stationären stochastischen Prozesse". In: *Mathematische Annalen* 109.1 (1934), pp. 604–615.

[22] Jong Wook Kim et al. "Crepe: A convolutional representation for pitch estimation". In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2018, pp. 161–165.

[23] Wilhelm Kirch. *Pearson's Correlation Coefficient*. Springer Netherlands, 2008, pp. 1090–1091.

[24] Anssi Klapuri and Manuel Davy. "Signal processing methods for music transcription". In: (2007).

[25] Yinmiao Li, Ziyue Piao, and Gus Xia. "A Wearable Haptic Interface for Breath Guidance in Vocal Training". In: *NIME 2021*. PubPub. 2021.

[26] Charles J Limb and Alexis T Roy. "Technological, biological, and acoustical constraints to music perception in cochlear implant users". In: *Hearing research* 308 (2014), pp. 13–26.

[27] James Mann. *Vocal Range Chart*. https://www.becomesingers.com/vocal-range/vocal-range-chart, 2022.

[28] Matthias Mauch and Simon Dixon. "pYIN: A fundamental frequency estimator using probabilistic threshold distributions". In: *2014 ieee international conference on acoustics, speech and signal processing (icassp)*. IEEE. 2014, pp. 659–663.

[29] Precision Microdrives. *Coin Vibration Motor*. https://www.precisionmicrodrives.com/coin-vibration-motors, 2021.

[30] Richard T Miyamoto et al. "Comparison of multichannel tactile aids and multichannel cochlear implants in children with profound hearing impairments." In: *The American journal of otology* 16.1 (1995), pp. 8–13.

[31] National Cancer Institute: SEER Training Models. *SS Model 14 - Melanoma*. https://web.archive.org/web/20090114060549/http://training.seer.cancer.gov/ss_module14_melanoma/images/illu_skin01.jpg, 2022.

[32] Francisco T. Moura. *Likert Scales: How to Use it to Measure Perceptions and Behaviors*. https://liveinnovation.org/likert-scales-how-to-use-it-to-measure-perceptions-and-behaviors/, 2020.

[33] A. Michael Noll. "Cepstrum pitch determination". In: *The journal of the acoustical society of America* 41.2 (1967), pp. 293–309.

[34] Scott David Novich. "Sound-to-touch sensory substitution and beyond". PhD thesis. Rice University, 2015.

[35] Sile O'Modhrain and R Brent Gillespie. "Once more, with feeling: Revisiting the role of touch in performer-instrument interaction". In: *Musical haptics*. Springer, Cham, 2018, pp. 11–27.

[36] Martin Piszczalski and Bernard A Galler. "Predicting musical pitch from component frequency ratios". In: *The Journal of the Acoustical Society of America* 66.3 (1979), pp. 710–720.

[37] Lawrence Rabiner and Biinghwang Juang. "An introduction to hidden Markov models". In: *ieee assp magazine* 3.1 (1986), pp. 4–16.

[38] Trish Rooney. "The Understanding of Contemporary Vocal Pedagogy and the Teaching Methods of Internationally Acclaimed Vocal Coaches". In: *International Journal of Learning, Teaching and Educational Research* 15.10 (2016), pp. 147–162.

[39] Rei Sakuragi et al. "CollarBeat: Whole Body Vibrotactile Presentation via the Collarbone to Enrich Music Listening Experience." In: *ICAT-EGVE*. Citeseer. 2015, pp. 141–146.

[40] V for Science. *Detecting pitch automatically - The intuition behind the YIN pitch detection algorithm*. https://www.youtube.com/watch?v=W585xR3bjLM&t=489s, 2021.

[41] Larry C Solberg, Linda P Fowler, and Virginia G Walker. "The use of an autochromatic tuner for the measurement of vocal fundamental frequency". In: *Journal of communication disorders* 24.1 (1991), pp. 51–58.

[42] Stephen M Stigler. "Francis Galton's account of the invention of correlation". In: *Statistical Science* (1989), pp. 73–79.

[43] Wei-Ho Tsai and Hsin-Chieh Lee. "An automated singing evaluation method for karaoke systems". In: *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2011, pp. 2428–2431.

[44] José A Vega et al. "The Meissner and Pacinian sensory corpuscles revisited new data from the last decade". In: *Microscopy research and technique* 72.4 (2009), pp. 299–309.

[45] Norbert Wiener. "Generalized harmonic analysis". In: *Acta mathematica* 55.1 (1930), pp. 117–258.

[46] Wikipedia. *Fundamental frequency*. https://en.wikipedia.org/wiki/Fundamental_frequency, 2012-current.

[47] Wikipedia. *Gooseneck (fixture)*. https://en.wikipedia.org/wiki/Gooseneck_(fixture), 2021.

Appendix A

Programming

A.0.1 Real-Time YIN Pitch Tracking Program

The program files used to provide visual pitch tracking used during evaluation can be accessed in the external appendix:
External appendix → YINTracker

A.0.2 YIN Algorithm

The YIN algorithm used for pitch tracking in SIRTAH can be accessed in the external appendix:
External appendix → Device Dependencies → src

A.0.3 code.py

The main program for SIRTAH operation is described below and can also be accessed in the external appendix:
External appendix → code.py

External dependencies are imported:

```
import array
import audiobusio
import board
import math
import pitch
import pwmio
import time

from sirtah.audioprocessing import remove_dc,
    normalized_rms, MovingAverage
from sirtah.noteprocessing import audioparams,
    get_pitch_difference, get_cvm_output_value
```

```
from ulab import numpy as np
```

The PDM MEMS microphone properties are established: pin GP19 is set as SCL (clock) and pin GP18 is set as SDA (data), sample rate is set in 'noteprocessing.py', bit depth is set to 16 bits and mono is set to True for single input. An array of 0s are to be generated in so that it can be filled with values generated by the microphone sampling data:

```
### OPERATION SECTION ###
def main():

    # Microphone Input
    mic = audiobusio.PDMIn(
        board.GP19,
        board.GP18,
        sample_rate = audioparams["sample_rate"],
        bit_depth = 16,
        mono = True
    )

    samples = array.array('H', [0] * audioparams["buffersize"])
```

The YIN pitch tracker is then initiated with a buffer size and sampling rate defined in 'noteprocessing.py'. The buffer length is set to '4', as it seemed to be fairly paced. The fundamental frequency moving average and coin vibration motor moving average are defined:

```
# Pitch Tracker, YIN Method
    pt = pitch.Yin(
        audioparams["buffersize"],  # Buffer
        audioparams["sample_rate"], # Sampling Rate
        0.15 #Threshold
    )

    BUFLEN = 4 #Buffer Length
    f0_mavg = MovingAverage(buflen=BUFLEN) # F0 Moving Average
    cvm_mavg = MovingAverage(buflen=BUFLEN) # Coin Vibration Motor Average
```

The PWM output is handled by pin GP17 with a vibration frequency set to 250 Hz (peak detection of the Pascinian mechanoreceptors) and the variable frequency set to 'True' because the vibration intensity will change (not be fixed). Initial PWM vibration output is set to zero:

```
# Output to Coin Vibration Motor
    cvm = pwmio.PWMOut(board.GP17, frequency = 250, variable_frequency = True)

    cvm_out = 0
```

In the loop, the microphone begins collecting samples. The samples are then sent to the pitch tracker to have their values estimated. If pitches are detected, they are processed. Otherwise, in the case of silence or inaudible sound input, no vibration will be output. The values processed by the YIN pitch estimator are then collected (then divided by two, due to

some anomalous doubling that occurs only when using the PDM microphone). The newly processed values are then smoothed through averaging:

```
while True:

        # Microphone 'Record'
        mic.record(samples, audioparams["buffersize"])

        # Preliminary Pitch Estimation
        f0_pre = pt.getPitch(samples)

        # Only picks up input if pitch found
        if int(f0_pre) == -1:
            f0_mavg.update(0)
            cvm_mavg.update(0)

            if int(cvm_mavg.get()) == 0:
                cvm.duty_cycle = 1

            continue

        # Begin Processing of Pitch
        f0_raw = f0_pre / 2

        # Update Moving Average Buffer
        f0_mavg.update(f0_raw)

        # Compute Moving Average
        f0_estimate = f0_mavg.get()
```

The frequency difference, index and target note values are processed and acquired through 'noteprocessing.py'. The values for the coin vibration motor are called (see 'noteprocessing.py' below for explanation) and used as output values for the PWM duty cycle:

```
# Obtain Frequency Difference, Index and Target Frequency
        fdiff, idx, target = get_pitch_difference(f0_estimate)

        # Obtain Value for Vibration Motor Output
        cvm_out_raw = get_cvm_output_value(fdiff, idx)
        cvm_mavg.update(cvm_out_raw)

        cvm_out = cvm_mavg.get()
        cvm.duty_cycle = int(cvm_out) if cvm_out > 0 else 1

        # Printing
        print("f0 est: ", f0_estimate)
        print(f"fdiff: {fdiff}, target: {target}")
        print(f"coin out: {cvm_out}")

        # End of Loop Updates
        time.sleep(0.01)
```

```
    #print("program done")

if __name__ == "__main__":
    main()
```

A.0.4 noteprocessing.py

The frequency comparison algorithm is described below and can also be accessed in the external appendix:
External appendix → noteprocessing.py

Dependencies are called and certain functions are exposed. Parameters for interpolation and audio are defined:

```
import math
from ulab import numpy as np

__all__ = [
        "freqs",
        "audioparams",
        "get_pitch_difference",
  "get_cvm_output_value"
]

def make_interp(lmin, lmax, rmin, rmax):
    ls = lmax - lmin
    rs = rmax - rmin
    scf = rs / ls
    def interp_fn(x):
        return rmin + (x-lmin)*scf
    return interp_fn

# Audio parameters
audioparams = {
        "sample_rate": 44100,
        "channels": 1,
        "buffersize": 512,
        "volume_thresh": 0.01
}
```

The frequency values (in hertz) for western music notation are pre-defined within an array. The generally acknowledged vocal range for singers has been defined as between E6 (bass) or 82.41 Hz and F6 (soprano) or 1318.51 Hz. Their neighboring notes bookend the array for threshold purposes. The frequency interval thresholds are the midpoint values between two notes. For example, the difference between frequency 87.31 and 82.41 is 4.9 Hz; divided in half it would be 2.45 Hz, so the midpoint frequency between these two notes is 84.86 Hz. The array is set up as the range between the first value and the second value, which is used to generate the amount of vibration (as described further below):

```
# Target Frequencies. C#/Db(2) to F6
target_freqs = np.array([
          77.78, 82.41, 87.31, 92.5, 98, 103.83,
    110, 116.54, 123.47, 130.81, 138.59, 146.83,
    155.56, 164.81, 174.61, 185, 196, 207.65,
    220, 233.08, 246.94, 261.63, 277.18, 293.66,
    311.13, 329.63, 349.23, 369.99, 392, 415.3,
    440, 466.16, 493.88, 523.25, 554.37, 587.33,
    622.25, 659.26, 698.46, 739.99, 783.99, 830.61,
    880, 932.33, 987.77, 1046.5, 1108.73, 1174.66,
    1244.51, 1318.51, 1396.91
])

# Intervals Between Traget Notes
freq_interval_tresholds = [
    (2.315, 2.45), (2.45, 2.595), (2.595, 2.75),
    (2.75, 2.915), (2.915, 3.085), (3.085, 3.27),
    (3.27, 3.465), (3.465, 3.67), (3.67, 3.89),
    (3.89, 4.12), (4.12, 4.365), (4.365, 4.625),
    (4.625, 4.9), (4.9, 5.195), (5.195, 5.5),
    (5.5, 5.825), (5.825, 6.175), (6.175, 6.54),
    (6.54, 6.93), (6.93, 7.345), (7.345, 7.775),
    (7.775, 8.24), (8.24, 8.735), (8.735, 9.25),
    (9.25, 9.8), (9.8, 10.38), (10.38, 11.005),
    (11.005, 11.65), (11.65, 12.35), (12.35, 13.08),
    (13.08, 13.86), (13.86, 14.685), (14.685, 15.56),
    (15.56, 16.48), (16.48, 17.46), (17.46, 18.505),
    (18.505, 19.6), (19.6, 20.765), (20.765, 22.0),
    (22.0, 23.31), (23.31, 24.695), (24.695, 26.165),
    (26.165, 27.72), (27.72, 29.365), (29.365, 31.115),
    (31.115, 32.965), (32.965, 34.925), (34.925, 37.0),
    (37.0, 38.84)
]
```

A logistic function scaler is used to provide a non-linear effect on the PWM output value associated to the vibration motor. The minimum and maximum duty cycle output values are defined, based on the amount of vibration that can be sensed through the device's 3D printed material:

```
# Logistic Slope(s) Control
SHARPNESS = 0.6 #d
PLATEAU   = 10  #c

# Coin Vibration Value(s); MIN, MAX
CVM_MIN = 20000
CVM_MAX = 65535

LO_SCALER = make_interp(-1, 0, CVM_MAX, CVM_MIN)
HI_SCALER = make_interp( 0, 1, CVM_MIN, CVM_MAX)

def logistic_interp(x, c=1, d=1):
```

```
    last_term = 1 / (1 + math.e**(c*d))
    if x >= 0:
        hi_term = 1 / (1 + math.e**(-c*(x-d)))
        return hi_term - last_term

    lo_term = -(1 / (1 + math.e**(-c*(-x-d))))
    return lo_term - last_term

def handle_inp_for_logfn(xp, idx):
    lo, hi = freq_interval_tresholds[idx]
    x = xp / lo if xp <= 0 else xp / hi
    return x
```

Due to numpy restrictions presented in using CircuitPython's ulab, a self-generated absolute value function is designed. The frequency difference is created from the absolute value of the difference between the pre-defined note frequencies and the pitch that is being picked up from the microphone. Indexes are generated from the minimum arguments of the frequency difference and the pre-defined frequencies:

```
def quick_abs(arr):
    return np.array([
        abs(arr[i]) for i in range(len(arr))
    ])

# Generating note and frequency value relationship
def get_pitch_difference(pitch):
    fdiff = quick_abs(target_freqs - pitch)
    idx = np.argmin(fdiff)
    desired_freq = target_freqs[idx]
    diff_to_return = pitch - desired_freq
    return diff_to_return, idx, desired_freq
```

The point that is furthest from a note or between two notes provides the maximum vibration output value, whereas the frequencies closest to a note will provide the minimum vibration output value. The values generated by the indexes are interpolated through the scalers and their values are used to define PWM values needed for the vibration output:

```
# [Note 1](Min CVM)<---->(Max CVM)<---->(Min CVM)[Note 2]
def logistic_interp_fn(freq_diff, idx):
    x = handle_inp_for_logfn(freq_diff, idx)
    xl = logistic_interp(x, c=PLATEAU, d=SHARPNESS)

    if xl <= 0:
        return LO_SCALER(xl)
    return HI_SCALER(xl)

def get_cvm_output_value(freq_diff, idx):
    return logistic_interp_fn(freq_diff, idx)
```

A.0.5 audioprocessing.py

The input signal processing is described below and can also be accessed in the external appendix:
External appendix → audioprocessing.py

Dependencies are called and certain functions are exposed:

```
import array
import math

from ulab import numpy as np
from ulab import utils

__all__ = [
        "get_volume",
    "remove_dc",
    "normalized_rms",
    "MovingAverage"
]

UINT16MAX = (2**16)//2
```

A 'Moving Average' class is designed in order to take several values generated by the YIN predictions of the input signal and average them out to create a smoother value. For example, if YIN produces something like [441.1, 440, 439.7, 440.8, 441.5, 439.9, 440.4...], it will average the output value to something closer to 440. This is useful in creating a less erratic output. Then, a 'get volume' function is used to sense whether there is any input or not within an audible decibel range by filling the sample array (elsewhere) with values other than 0:

```
class MovingAverage():

    def __init__(self, buflen):
        self.buflen = buflen
        self.buffer = [
            0 for _ in range(self.buflen)
        ]

    def __rollbuf(self, value):
        for i in range(self.buflen - 1):
            self.buffer[i + 1] = self.buffer[i]
        self.buffer[0] = value


    def update(self, value):
        self.__rollbuf(value)

    def get(self):
        return sum(self.buffer) / self.buflen
```

```
def get_volume(samples):
        return np.sum(samples**2)/len(samples) * 100
```

This section of the code is commonly used when utilizing the PDM MEMS microphone. The square root of the mean (RMS) of the samples captured by the microphone is normalized in order to return the microphone levels. As indicated in the preceding comment, the 'mean' function generates the mean or average of the microphone levels and those values are used in order to remove and direct current bias. These functions quickly capture multiple sound samples and average them to generate a more accurate value from the input signal:

```
# Return Microphone Levels
def normalized_rms(values):
    minbuf = int(mean(values))
    samples_sum = sum(
        float(sample - minbuf) * (sample - minbuf)
        for sample in values)

    return math.sqrt(samples_sum / len(values))

# Average Microphone Levels/Remove DC Bias
def mean(values):
    return sum(values) // len(values)

def remove_dc(values, inplace=True):
    return values - np.mean(values)
```

Appendix B

Long-Term Post-Evaluation Interview

B.0.1 Participant 1

If they had a month to use the device, they would be able to make better use of the 'muscle memory' aspect of the vibration feedback. The device could be useful as an initiator to practicing singing; using it to know what note is in pitch. The learning curve might create frustration because of the desire to correct the amount of vibration, but getting over the frustration could happen after extended use. The device was easy to use. They would have appreciated knowing that the length of the feedback was equivalent to the duration of the sung note. Also, they noticed on a physiological level that something felt right when the pitch was sung accurately, including how they used their core to sing. Felt more confident with the device than singing solo. Had less time to use the device due to illness.

B.0.2 Participant 2

They mentioned that it worked well as a guiding device, where they could feel how the sung note felt correct when they adjusted their voice based on the vibratory feedback. At first, it was difficult to understand or 'appease' the vibration of the device, but after time, they got used to it and realized what they needed to do to adjust their pitch slightly. They also realized they had to sing louder or move their head to the side for the microphone to pick up the signal. They thought this could be useful in encouraging singing projection (singing louder).

B.0.3 Participant 3

They suggest it would probably be more accurate than playing piano notes. They thought the feedback was very good. Helped to train singing sustained notes. After a few days, they began to notice improvement in singing. Seemingly, the only confusion was if the device was working, due to the requirement to sing

a little louder and having no vibration when not picking up a signal. When it was apparent that the device was working and was used, they began to notice improvement. Having to use the device gave motivation to practice singing. For longer use, the device could be used to train hearing [of how musical pitches sound].

B.0.4 Participant 4

They did not know whether the vibration was supposed to be minimized or stop vibrating. The experience probably would have been better if the device was used more consistently. They paid more attention to whether the sung notes were in tune or not. The device was always vibrating a little bit (could have been a recent fault in the mechanics). Because the device required slightly louder singing, and because they sang more quietly, it was an adjustment to try and sing louder. Also, because they generally do not consider themselves a singer, understanding the techniques within themselves required more effort.

B.0.5 Participant 5

They felt the device was somewhat uncomfortable when worn. A visual aid could be more effective in correspondence with the vibration. The vibration method was somewhat more difficult to understand and was not perceived as useful. They would personally not use the device but they think that the device should be tested with other hard-of-hearing individuals that are learning to or wanting to improve their singing.

B.0.6 Participant 6

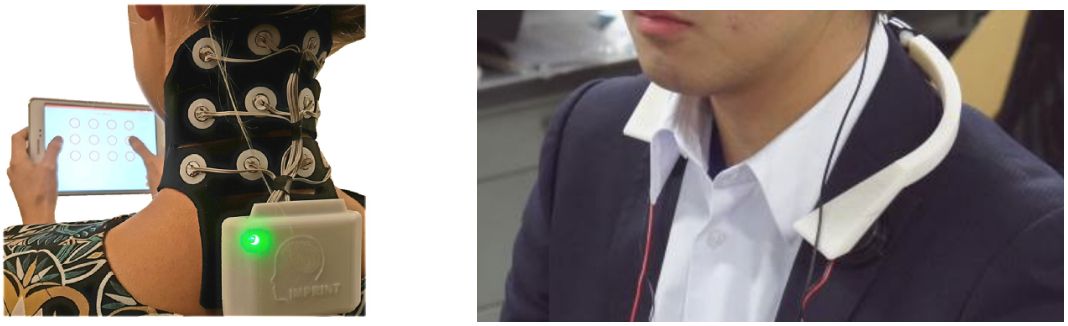
They think the idea of the device is very nice. However, there is no direction to the device (telling you whether to adjust voice up or down). The chromatic scale is closer together in frequency, so they suggest that having a pentatonic scale (for example) could provide a greater gradient between right and wrong sung pitches. They noticed there was a small amount of latency that led to not being sure whether the device was working correctly or not, and when the device is not vibrating, it could mean that it is either picking something up that is on pitch or when no signal is detected. They think this could cause confusion and, therefore, maybe mapping the feedback to vibrating when you are singing in pitch might be more effective and it might also be good to vibrate at the frequency of the nearest note. They suggest there are a few more steps to go for it to truly work.

Appendix C

Images for Additional Technology

Here are the additional images of the technologies described in section 3.2 and chapter 2.

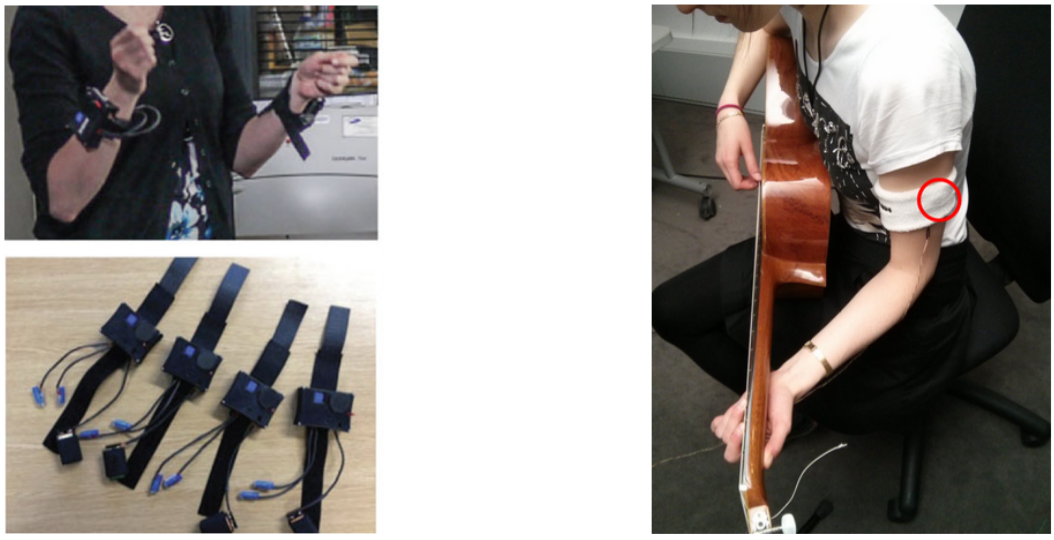
C.1 Perception Studies



(a) **(b)**

Figure C.1: (a) The haptics device used in the localization test mentioned in section 3.2 [17]. **(b)** The Collarbeat, as used for collarbone conduction in section 3.2 [39].

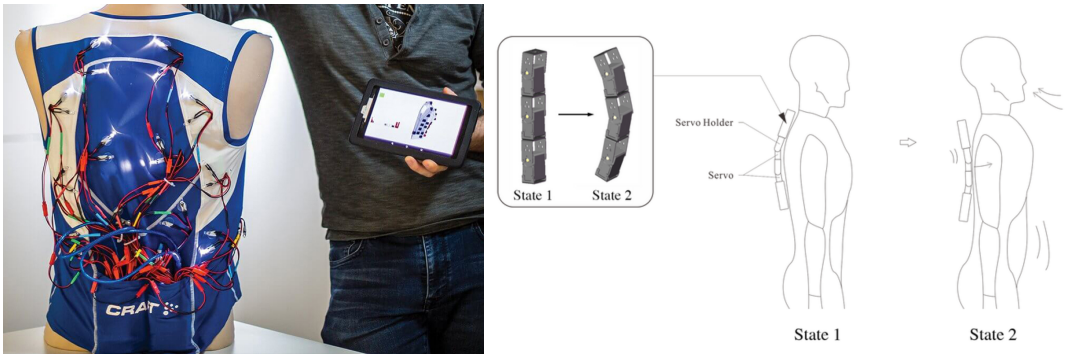
C.2 Music Haptics



(a) (b)

Figure C.2: (a) The haptics bracelets as mentioned in section 2.2.1 [5, 20]. **(b)** A guitarist wearing an armband with built-in vibration motor as mentioned in section 2.2.2 [13].

C.3 Additional Similar Technologies



(a) (b)

Figure C.3: (a) The VEST as mentioned in section 2.3.1 [34]. **(b)** The spine haptics device to help with breathing as described in section 2.3.2 [25].

Appendix D

Additional Materials

D.0.1 3D Models

The 3D models for the SIRTAH component enclosures that were designed in Tinkercad can be accessed in the external appendix:
External Appendix → 3D Models

D.0.2 Microcontroller Build

The .u2f file used to program the microcontroller can be accessed in the external appendix:
External Appendix → Device Dependencies → build → sirtah_firmware_RP2040.uf2

D.0.3 Consent Form

The consent form the short-term participants were asked to sign can be accessed in the external appendix:
External Appendix → Evaluations → Consent Form - Vocal Device.pdf

D.0.4 Audio Recordings

The edited recordings of all participants can be accessed in the external appendix:
External Appendix → Evaluations → Recordings

D.0.5 Pitch Accuracy Script

The MATLAB script used to determine pitch accuracy can be accessed in the external appendix:
External Appendix → Evaluations → audioplot.m

D.0.6 Pitch Accuracy Plots

The MATLAB plot graphics used to determine pitch accuracy can be accessed in the external appendix:
External Appendix → Evaluations → Pitch Accuracy

D.0.7 Pitch Accuracy Spreadsheet

The spreadsheets used to collect pitch accuracy data can be accessed in the external appendix:
External Appendix → Evaluations → Pitch Accuracy

D.0.8 SIRTAH SUS Spreadsheet

The spreadsheets used to collect System Usability Scores can be accessed in the external appendix:
External Appendix → Evaluations → Results

D.0.9 Friedman's ANOVA Script

The script written using Friedman's ANOVA on the short-term evaluation data can be accessed in the external appendix:
External Appendix → Evaluations → ANOVA