Titel

Project Report

Group MTA 16440

Aalborg University
Media Technology
Rendsburggade 14
DK-9000 Aalborg

AALBORG UNIVERSITY
STUDENT REPORT

Media Technology
Rendsburggade 14
DK-9000 Aalborg

Title:
Title

Theme:
Sound Computing and Sensor Technology

Project Period:
Spring Semester 2016

Project Group:
MTA 16440

Participant(s):
Alex Bo Mikkelsen
Allan Schjørring
Daniel Agerholm Johansen
Didzis Gailitis
Liv Arleth
Sebastian Laczek Nielsen

Supervisor(s):
Olivier Lartillot

Copies: ??

Number of Pages: 82

Date of Completion:
May 26, 2016

Abstract:

Abstract

Contents

Preface

Aalborg University, May 26, 2016

Alex Bo Mikkelsen
<amikke13@student.aau.dk>

Allan Schjørring
<aschja14@student.aau.dk>

Daniel Agerholm Johansen
<djohan14@student.aau.dk>

Didzis Gailitis
<dgaili14@student.aau.dk>

Liv Arleth
<larlet14@student.aau.dk>

Sebastian Laczek Nielsen
<slni14@student.aau.dk>

1. Introduction

1.1 Initial Problem Statement

The problem addressed in this project deals with real-time access to voice effects for performers. Some performers finds it useful to use voice effects while performing. This project focuses on making the effects accessible and interchangeable in real-time. Instead of most commonly only being applied in set amounts by the push of a button, it could be beneficial for the performer to be able to tamper with the effect parameters in real-time also. In many cases it is not the performer that controls which effects are applied and when, rather it is applied by someone backstage or someone else.

Additionally, many existing devices used for adding voice effects can be difficult to use, say during a concert. This could be a pedal, which often just has set amounts of effects, where the performer pushes a button and e.g. a harmony is applied. The problem with the pedal is that it is stationary and the performer will have to reach for it to apply an effect, which is inconvenient in some cases.

In this project, it is the desire to give the performer full control of the effects and its parameters, while moving freely across the stage.

It could be interesting to make a wearable device that grants the performer control of the effects, while also being intuitive for the performer to use.

1.1.1 Statement

In order to obtain the necessary information regarding the subject, the following statement has been formed.

How does one create a wearable device that applies voice effects to a voice in real-time, while also having an intuitive interface design?

1.1.2 Target Group

The criteria for the target group in this project are:

- They should have experience with performing

The target group consists of performers that know about voice effects. They should not play an instrument or similar while performing, because they must be able to use their body for controlling the effects. There is no specific genre or type of performer as the only criteria is that they know the technicalities behind performing.

People who fulfil these criteria could be singers, stage actors and stand-up comedians.

2. Problem Analysis

This chapter focuses on the research necessary for the basic understanding of the subject. The focus lies on answering the following research questions:

- What are the most common voice effects?
  - Which of these effects would performers have the desire to change during a performance?
- Does any existing technology use body gestures or sensors to apply effects?
- Which gesture should be used with which effect, and how?

The research will include a short description of effects that could be applied by the system, the current state of the art within the field, and some theory behind the use of gestures.

2.1 Effects

Many voice effects exist today. Some effects are used by singers and some are used by other performers. The effects can be really subtle, or really noticeable. In this section, there will be short descriptions of some common effects.

2.1.1 Delay

A delay effect creates a repetition of the original sound after a period of time[1]. By using the delay effect, it is possible to simulate the sound of the echo created when yelling into a cave or over a canyon, which some people may find desirable to use in their song.

2.1.2 Reverberation

When sound reflects off surfaces in a confined space, its called natural reverberation[2]. Reverberation like this works best when the sound hits hard surfaces. For example, the sound effect that comes when you sing or yell in a church, is reverberation. The sound bounces all around the church's hard walls. Digitally, the way to simulate reverberation is to use a multitude of delays and feedback. This then creates a series of echoes that then slowly decays. If the amount of reverberation is small, it can simulate singing in a small or medium sized room, even though the voice input has no natural reverbaration. If the amount of reverberation is big, it can make the voice sound like singing in a church, even though you are not in a church. This can make a person's voice sound "big".

2.1.3 Pitch Shift

The frequency of a harmonic sound is called its pitch[3]. By shifting the pitch, the sound will effectively become deeper or higher. An example of this is the voice that anonymous people get when they want to hide their voice, this is a lowered pitch. Another example is the "chipmunk voice", which is achieved through a raised pitch. Pitch shifting can be done by using the "phase vocoder", which is a digital signal processing technique[4]. The phase vocoder works by analysis and synthesis. The analysis part takes the signal, and models it as a sine wave in which one can find the amplitude, phase, and frequency of the sine wave. In the synthesis part, one can manipulate these parameters. The phase vocoder can do many things, e.g. change the pitch of a sound without changing the duration of the sound - make a sound deeper or higher in real time. People use pitchshifting to make their voice deeper or higher. People want to make their voice deeper to e.g. simulate a bass guitar, and make their voice higher to achieve e.g. a "chipmunk" voice.

Pitch shifting is also used to create the harmonizer effect. It takes the input voice and shifts its pitch a certain amount (in relation to half tones), and then adds it as an additional voice. This can effectively simulate a choir.

2.1.4 Auto-Tune

The Auto-tune effect corrects a singer's voice to the correct tone[5]. This can be really subtle or plainly obvious. The user just needs to choose a reference of scales or tones and the amount of correction that needs to be made, and then the Auto-tune will make the proper adjustments.

2.1.5 Vocoder

The Vocoder effect combines a performer's voice with another sound - that could be the sound from an instrument or a synthesizer[6]. The effect can make the voice sound like a robot. The vocoder needs two inputs, the voice and e.g. an instrument. The fundamental frequencies of the voice are converted to levels of amplitude on a series of band pass filters, which then are passed through the instrument sound.

2.1.6 Discussion

Changing effect parameters can benefit some effects, but not all. In a regular reverb effect, one can change the amount of reverb to be added, add high or low-pass filters. Changing reverb parameters would probably be too subtle for the untrained to notice. The delay effect has the parameters: amount of feedback, length of delay, and some kind of filter. The amount of feedback and length of delay are parameters to consider, since it would be easy to hear a difference in. Pitch shifting is also a good effect to change parameters in, in this effect it would be to change between high and low pitch. The harmoniser uses multiple pitch shifters, but the parameters one could change would be the amount of harmonise effect, and degree of pitch shift (how many halfsteps to pitch shift). The harmoniser is also a good effect, since it is also easy to hear a difference in. The vocoder effect's parameters would be the amount of effect, but since it needs a instrument and voice input, it is not an effect to consider. The autotune effect's parameters are amount of correction, and which tones it corrects to. This effect could be considered because it will also be possible to hear a difference in.

2.2 State of the Art

To gain understanding of what is possible, a study of the state of the art was conducted with the focus on commercial artefacts used for real-time alterations.

2.2.1 TC Helicon Perform V

The TC Helicon - Perform V is a vocal multi-effects processor that attaches to a microphone stand, as seen in figure 2.1[7]. It has three effect buttons, three preset buttons, a big knob, and other buttons. The effects are reverb, echo, "double" (harmonizer), equalizer, compressor, and many more. It is possible to download an application that can connect with the Perform V. The application has many premade sounds, and it has a wireless connection.

Figure 2.1: TC Helicon Perform V[7]

The Perform V is good for live performing if the singer has the processor in front of them, on the microphone stand. Preset buttons make it easy to change effect quickly. If the singer plays an instrument, it is probably difficult to change effects without interrupting the instrument playing. Another downside is that the singer is limited to only three presets, and only one knob to turn.

2.2.2 Electro Harmonix Voice Box

The Electro Harmonix Voice Box is a more advanced processor than the TC Helicon[8]. It has six knobs: blend, two reverb knobs, “gender bender”, voice mix, and “Mode”, as seen in figure 2.2. It has nine different modes, which includes different kinds of harmonies, unison-whistle, and a vocoder, which the TC Helicon does not have.

Figure 2.2: Electro Harmonix Voice Box[8]

The Voice Box has to be on a flat surface, like the floor or a table and is most often used as a pedal. It is possible to insert an instrument into the pedal, so it can be used for the vocoder. The Voice Box has many effects and knobs - this can make changing effects and effect parameters difficult, even more if the pedal is on the floor.

2.2.3 Mi.Mu Gloves

The Mi.Mu Gloves are gloves made for making music, and controlling sound[9]. They are made by scientists, musicians, and artists, and have been in development since 2010. They are wearable, and can be used by one or both hands, see figure 2.3. The gloves have been through many iterations, and they are open source. The gloves use gestures, hand and finger movement, finger placement, and other features to control sounds and effects. The hardware includes an ArduImu, flex/bend sensors, accelerometer, gyroscope, haptic motors, LED's, Wi-Fi compatibility, and provides other capabilities.

Figure 2.3: Mi.Mu Gloves[9]

The gloves are bluetooth or Wi-Fi connected, so the person using the gloves are free to move around, and does not have to worry about wires. They are also battery powered. Since the gloves are open source, you can make your own - many different gloves exist - some are simple, and some are complex.

2.2.4 HandySinger: Expressive Singing Voice Morphing using Personified Hand-puppet Interface

Yonezawa et al. made a glove that controls voice effects [10]. The wearer of the glove controls a puppet and and makes hand gestures, see figure 2.4.

Figure 2.4: HandySinger Glove [10]

They believe that using a puppet interface will increase the expressiveness of the user's singing voice. The glove itself has seven bend sensors, and two pressure sensors, see figure 2.5.

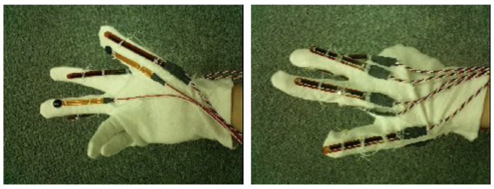

Figure 2.5: HandySinger Glove sensors [10]

The glove measures both forward bend and backwards bend. The gestures that the users can make are: bend back clasp, drooping, stretching, and bend back. The parameters that the gestures change are: "dark", "whisper", "wet", and volume. Yonezawa et al. found that users with small hands had trouble using the glove effectively. Nevertheless, they confirmed it was easy to gesture with the hand-puppet and that the gestures reflect the voice expression changes.

2.2.5 The 'One-Person Choir': A Multidisciplinary Approach to the Development of an Embodied Human-Computer Interface

The study by, Maes et al. [11] utilises body gesture to enhance a singer's voice. The system is a human-computer interface that use gestural control for harmonising a singing voice. The system is operating in real-time, which means it is possible to use it during live performances. The system can be seen in use in the figure2.6. The system uses pre-configured models to control the harmonisation, and the singer can eventually use this to enhance his or her singing voice. During their research, they found that gesture control is a big part of singing, which also helps the perception of the singing. The movement of the upper body is the primary gesture used in the system which means that the singer has sensors attached to the upper body.

Figure 2.6: The One-Person Choir in use[11]

2.2.6 Discussion

Researching the state of the art has shown, that there exist different kinds of products that all are able to manipulate a performer's voice. There exist many different pedals, like the Electro Harmonix Voice Box, that can change effects, but such pedals has to be on the ground out of reach[8]. The Voice Box has many effects and knobs, but this means it might be difficult to change effects and effect parameters quickly, because there are so many. On the other hand there is the TC Helicon Perform V, which enables voice manipulation from an easier to reach point - the microphone stand[7]. The Perform V has few, but big buttons and a big knob to turn, and this means the singer can change effects faster than the Voice Box. In order to change effects in both effect units, one has to look at the effect unit and find the buttons and knobs to turn, in order to change effect or effect parameter. The Mi.Mu gloves shows how it is possible to enable a performer to manipulate a voice and create different sounds, while moving around freely[9]. Studies by Yonezawa et al. and Maes et al. shows that gesture control is a big part of singing[10][11]. Combining gesture control with the concept of the Mi.Mu glove could be interesting.

2.3 Gestures

When designing a way to control effects, there are several ways to approach it. One of these ways is through hand gesture control.

There are three stages of a gesture: registration, continuation, and termination, as seen in figure 2.7[12, pp. 127-134].

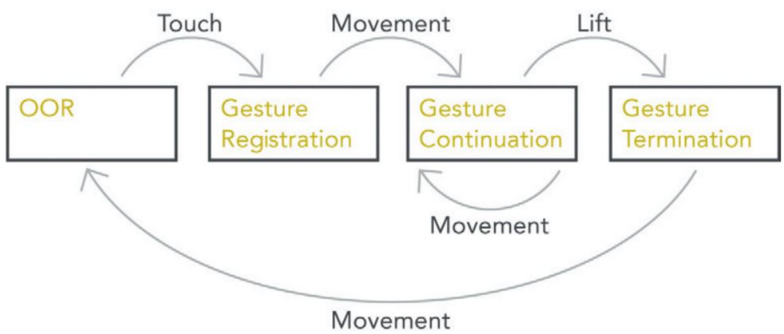

Figure 2.7: Out of Range(OOR) and the three stages of gestural input [12, pp. 127]

The registration stage is when the system registers that the user would want to perform a gestural input, e.g. when you place your finger(s) on a touch display.

The next stage is continuation. In this stage, the user uses movement to adjust the parameters of the gesture, e.g. when you move two fingers away from each other in order to zoom, on a touch display.

The final stage is termination, where the user simply ends the gesture, e.g. by lifting their finger(s) from a touch display.

In some cases the continuation stage can be skipped. Whether or not it is a good idea to skip this stage is decided by whether they want the gesture to be able to change the parameters or not, e.g. a zoom that does not have one set amount of zoom, but rather it is specified by how much you move your fingers. By removing the continuation stage, it also removes the possibility to differentiate between a lot of gestures.

A good thing to do, when choosing a gesture for a specific action, is to keep it as unambiguous as possible. This helps reduce future errors. This being said, it is also important to minimise the amount of steps the user has to go through for the gesture.

While the previously mentioned 'zoom' gesture is a rather good example of a gesture used to change parameters, a bad example of such a gesture would be the use of the 'flick' action to execute a gesture. The 'flick' action is when you execute an action by 'flicking' an object to e.g. delete it. The bad thing about this is that it implements no way to change the parameters dynamically. Another bad thing about it is that one has to specify a border between the action of simply moving something across the screen and 'flicking' something on the screen.

2.3.1 Discussion

The desire of this project is to not only apply effects, but also be able to change the parameters dynamically during a performance. With this in mind, it is a requirement to include all three stages to make a fitting gesture for the different effects, since including the continuation stage is the only way to make this possible.

2.4 Conclusion

There are numerous sound effects each unique in the way that it works and is applied, so understanding them is important as each of them has different limitations or applications in a real-time setting. By going through each of them and learning about them, the effects most relevant to the purpose of the project was picked and described.

From the state of the art it can be seen that extensive effort has been put into creating many different devices of varying complexity and limitations. In design most of them vary between a glove and a plug in device, those design differences also provide different accessibility while performing. Additionally, the state of the art provided information about which voice effects professional singers would usually like to adjust and how.

Finally, by researching what a proper gestural input should consist of and how it should be performed, it was discovered that there are several stages to a gesture, and some of the stages are more important than others, depending on the purpose of your gesture. Furthermore, when looking at the previous state of the art examples, it can be concluded that two of the most common gestures for a singer to apply an effect and change its parameters, is by pushing a button and to turn a knob, as can be seen in figures 2.1 and 2.2.

2.5 Problem Statement

Based on the findings from the research a final problem statement has been made:

How can one design and implement a wearable device with an intuitive interface that applies voice effects in real time using gestures, without the restraints of existing solutions?

2.5.1 Success Criteria

Based on the problem statement, some success critera has been made which are as follows:

- The system should have at least two effects
- Users use the correct gestures to change the effects
- The system performs the correct effect corresponding to the gesture, meaning it does not misinterpret

The system should have at least two effects to make it possible to test different effect parameters. It is important that the users perform the gesture that corresponds to the effect, since the evaluation is based on these effects working. It is also important that the system performs the right effect, whenever a gesture is performed. As an example, whenever a user performs the knob turning gesture, the system has to perform the effect that matches the gesture.

2.5.2 Minimum Implementation Requirements

The following describes the minimum implementation, which are needed for the prototype to be testable. These are specified by us. The minimum requirements are as follows:

- The system must implement the use of an Arduino
- The system must implement at least one sensor applicable to the Arduino
- The system must implement audio processing
- The system must be implemented using the Arduino software and PD Extended
- The system must get audio from a microphone

3. Design

This chapter describes the design process for this project. Initially, the concept of the product is presented with the storyboard to explain the product of the project. The product is then shown and described with the sketches and the iterations that the product went through in the project. Next a walkthrough of the mental model lo-fi test together with an affordance scheme and the results of the test. In the end of the chapter, the design choices for the product are discussed and a conclusion for the chapter.

3.1 Concept

The concept of the product is to be able to apply voice effects in real-time without using a panel or have someone backstage do it for you. The idea is based on giving the singer/performer more freedom and control over how they want to sound in a natural way, compared to turning knobs or using sliders. Most singers always have their hands available, therefore the concept is based on a controller based on hand movement. The hand movement would be different gestures to control which effect to apply and the degree of it. The controller would implement a gyroscope to sense the hand and wrist movement, the functionality of the gyroscope is explained further in the implementation chapter. The chosen gesture is based on which fingers the user connects. The idea behind the use of two fingers is to create a connection that initiates a specific effect. The index- or middle finger each control their own effect when the finger is connected to the thumb. The two original ideas for voice effects were pitch shift and reverb. Reverb was later changed to harmonisation beacuse it is easier to hear a parameter change in.

- Harmonising: This will be controlled by turning the hand while having thumb and a finger pressed together, like turning a knob or volume control.
- Pitch: This will be controlled by lifting or lowering the hand while having thumb and a finger pressed together, like pulling a slider up or down.

3.2 Storyboard

To understand the concept and how it would work in a real scenario a storyboard was created to explain how, what and where to use the product, see figure 3.1. this storyboard was part of the proof of concept that together with the initial sketches would explain the idea better.

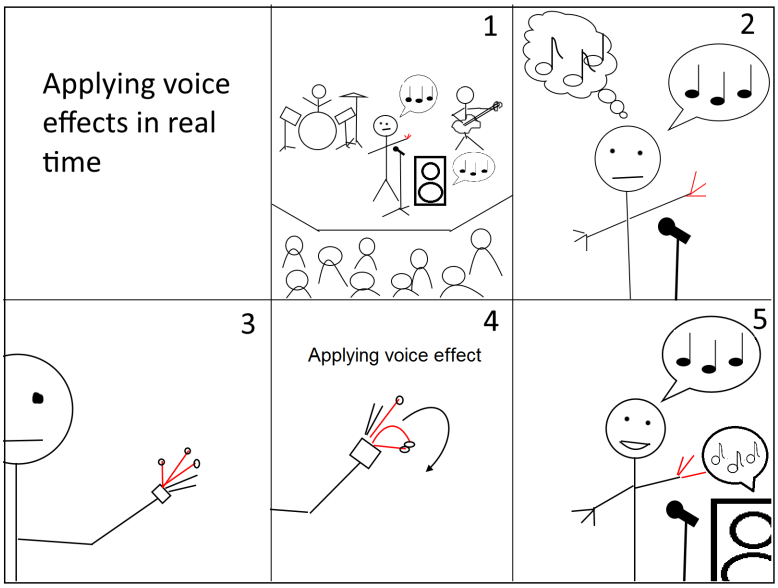

Figure 3.1: The Storyboard

In the first frame a band is shown playing a concert on a stage, with the lead singer singing. The second frame shows the lead singer singing, but wanting to sound different. In the third frame the device is shown on the singer's hand. Fourth frame illustrates the singer performing a gesture to apply the desired effect. The final frame then shows the singer sounding how they wanted to sound.

3.3 Sketching and Testing

The initial sketches and the first concrete design of the device have copper foil on the thumb, index finger and middle finger, see figure 3.2. When pressing the index or middle finger together with the thumb, a connection is made in the system that activates the assigned effect.

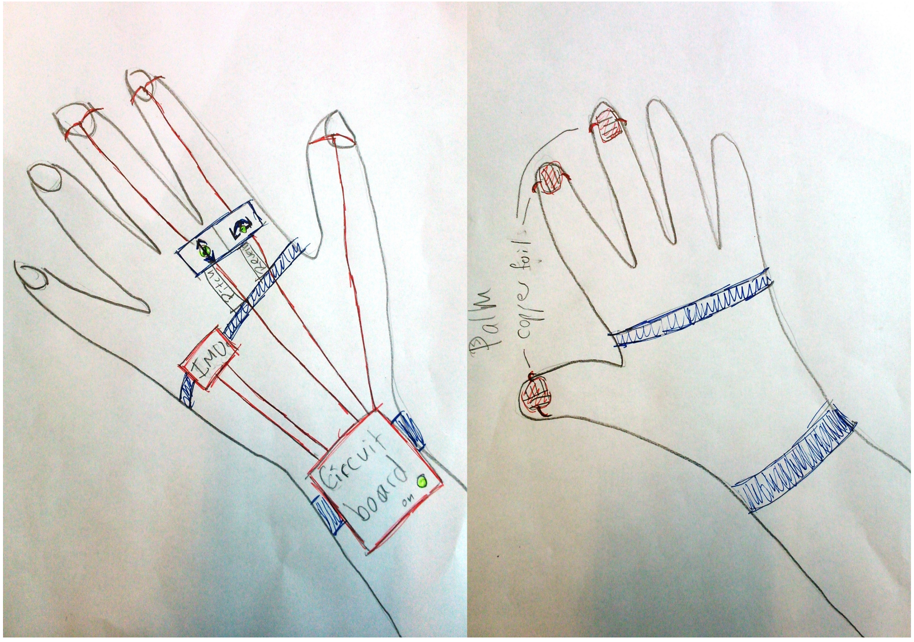

Figure 3.2: The initial sketches showing the idea of the system

On the knuckles there are illustrations of the gestures, that you are supposed to do to manipulate the effects. Beneath those are small labels with the name of the effect. In this design reverb is used instead of harmonising. This was later changed, since reverb is not manipulated quite as much as harmonising.

The sensor is attached to the hand by a velcro strip, as is the circuit board.

A quick informal test with three participants was conducted and they were told what the drawing was supposed to be and what it should do. They then had to figure out based on the sketch how to do those things. From this test, a few key things were learned:

- They all had difficulty figuring out how to get to the activate stage. None of them connected their fingers.
- Most eventually figured out which type of gesture in general had to be done, but only after some trial and error
- They all found out which finger created which effect

The second sketch changed the illustrations since people had a hard time immediately recognising the correct gestures with the old ones, as seen in figure 3.3.

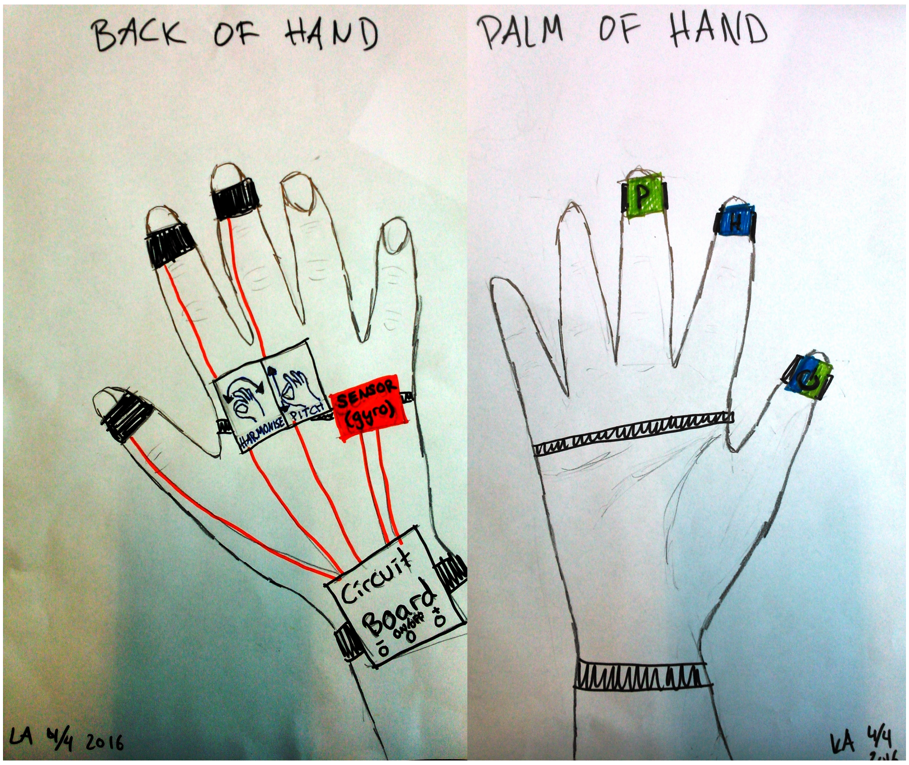

Figure 3.3: Second set of sketches after some changes

Colour was also added to the copper foil, a different one on the index and middle finger and then both on the thumb. This was done to create perceived affordance between the fingers and thumb.

LEDs were added on the circuit board to create some feedback for the actions. The middle LED showing if the system was on or off. The two other LEDs showing an increase or decrease in the effect, with a minus and a plus sign.

A quick informal test was done with two participants with the revised sketch. Now there was a better indication that they needed to connect two fingers, but not anything that indicated that they needed to stay connected. Additional results from the test:

- One suggested that instead of the on/off LED, maybe a connected/not connected LED.
- The arrows were found to be confusing for one tester.
- Another tester easily understood the pitch action but was a bit confused with the placement of the arrow on the harmonise action.
- One tester thought tht the dual colour on the thumb suggested that both actions could be done at the same time.

- The plus and minus LEDs confused one tester, but this could also be because the drawing was unclear.

3.4 Lo-Fi

Based on the previous test, a new design iteration of the glove was made, now also in the form of a lo-fi model, as seen in figure 3.4. This lo-fi was created in order to make a mental model test. The desire was to both get some feedback on the general design, but also to weigh the users' mental model against an affordance scheme.

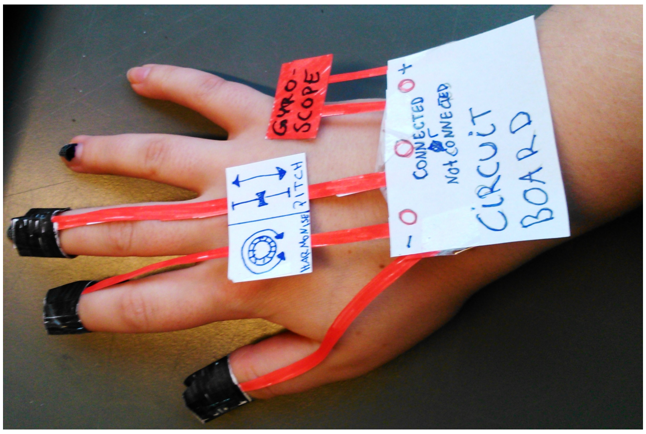

Figure 3.4: The Lo-fi of the system

3.4.1 Affordance Scheme

The affordance scheme is separated into three categories: perceived affordance, feed-forward, and feedback[13]. Perceived affordance is the perception that something is interactable, e.g. a user would assume a button can be pressed no matter what state the system is in. Feedforward is what is expected to happen after a certain action, e.g. after pressing the "On" button, the system will turn On. Finally, feedback is what the system does to indicate an action has taken place, e.g. after pressing the "On" button, a display lights up and says "Turning On".

The following table shows the affordance scheme for the system.

State	Perceived Affordance	Feedforward	Feedback
Inactive	It is a wearable glove		All LEDs are OFF
Active		Connect fingers + perform gesture on labels = effect change	"Connected" LED is ON
Perform-ing Gesture		Connect fingers + perform gesture = effect change	Voice Changes and the plus and minus LEDs light accordingly

3.4.2 The Test

The test was initiated with a short introduction for the user, without revealing too much. Then the participants were asked to explain everything they saw and assumed about the glove, and to try to put it on. Following this, they were given the tasks of turning harmonisation up and pitch down. Finally, some follow-up questions were asked to figure out why they did what they did.

Seven people participated in the test. Here are some results from the test:

- When trying to apply the effects, 6 out of 7 testers' first reaction was to not connect their fingers
- 5 out of 7 did the correct gestures, based on the ilustrations on the knuckles
  - 1 of the two who did not do the correct gestures, did what she did because she thought the illustrations were interactable
- 6 out of 7 found the illustrations helpful
- All participants figured out that the device was a glove to wear on the hand
  - 5 out of 7 put it on correctly, as seen in figure 3.4
- All participants were in doubt of what the LEDs were for

The table below shows a revised affordance scheme based on the user feedback. The parts of the system that the participants had a hard time understanding are shown in red writing in the table. Additionally, the parts which were generally understood by the users has the text coloured green.

State	Perceived Affordance	Feedforward	Feedback
Inactive	It is a wearable glove		All LEDs are OFF
Active		Connect fingers + perform gesture on labels = effect change	"Connected" LED is ON
Performing Gesture		Connect fingers + perform gesture = effect change	Voice Changes and the plus and minus LEDs light accordingly

The participants had a hard time understanding some of the feedback in the system, particularly the LEDs. While they knew that they were LEDs, they were not sure what each of them was signifying. This could also be a result of bad simulation of the LED state changes in the test. Another thing that the users had a hard time understanding was the colour coding on the fingers. Most did not connect their fingers to initiate the effect change.

The things that the users perceived correctly was that the system was to be used as a glove. After the alterations based off of earlier tests, the new illustrations had much more success regarding the gestures. Most understood the feedforward provided by the labels and performed the gesture correctly, although without connecting the fingers.

3.5 Design Choices

Throughout the design process a lot of choices have been made regarding the design of the system. The reasoning behind these decisions will be discussed in this section.

Very early on in the process, it was decided to make the gestures base off of some gestures that people would normally associate with manipulating music, like a volume knob or a slider on an equalizer, as seen in figure 3.5.

Figure 3.5: The AX 301 Synthesizer Amplifier with knobs and sliders[14]

According to Wigdor and Wixon it is important to mimick real world physical interactions to create more natural experiences for the user[12, pp. 47]. The gesture of turning a knob was assigned to the harmonize effect, while the vertical slider gesture was assigned to the pitch shift. Since pitch shift is manipulated by making a higher or lower pitch, it makes sense to make the gestures simulate a vertical slider, with a higher pitch being assigned to moving up, and lower pitch assigned to moving down. Furthermore, in accordance to the "turning a knob" and "moving a slider" sentiment, it had to be implemented on the actual glove. This was done by adding some copper band on some velcro for the fingers. This copper was then connected to the circuit board with wires. When connecting the fingers, this would then allow the corresponding effect to be activated, and then the effect could be changed by using the gestures.

When looking at the feedback for the glove, other than the voice changing, it was decided to use LEDs as a way to communicate state changes. A red LED was used to indicate that the effects were being changed in the negative direction, e.g. pitch being lowered, and a green LED was used for the positive direction. Additionally, these LEDs would be accompanied by labels. A minus sign for the negative direction and a plus sign for the positive. The red LED and the minus sign would be placed on the left side of the circuit board to indicate the negative direction the "knob" had to move. The green LED and the plus sign would be placed on the right side to indicate the positive direction. Finally, a yellow LED was placed in the center of the circuit board. This LED will indicate whether the glove was active or not. Active being the state where an effect could be changed.

3.6 Conclusion

The first concept of the system was a glove that uses a gyroscope to sense hand movement, and apply voice filters according to the movement. A storyboard was created to show how the system would be used and in what context. The first sketch showcased the system with copper foils on the fingers, and labels to explain how to operate the device. This sketch included reverb as an effect to apply, which was later changed since reverb is mostly irrelevant for a singer. The second iteration also included new labels to provide better feedforward and LEDs for feedback when interacting with the system. The third iteration once again included new labels, this time with more success and was made into a lo-fi model. This iteration was compared to the affordance scheme with the results showing the labels were better, but the system lacking elsewhere, namely an indication of the connection of the fingers to apply an effect. The test participants also had some trouble understanding the feedback of the system, as in how the LEDs worked and what sort of feedback they actually provided.

4. Theory

This section explains the theory behind the effects that has been used in the implementation.

4.0.1 Sampling

Sampling and quantizing is used as a way to convert analog signals into digital. Sampling is discretizing the time variable. Meaning splitting the continuous signal into discrete signals, by isolating values at regularly spaced time intervals. Quantizing is the same operation but done for the pressure signal, which represents the displacement of microphones diagram from its resting place, these value range from -1 to 1.

When the signals have been converted into discrete signals they can be split up into samples. Sample being a value or sometimes a set of values in time. The average amount of samples that each second is split up into depends on the goal and use of the signal. This is called the sampling rate/frequency(Hz) and generally speaking the higher the sampling rate the wider range of sounds can be heard and the signal is higher quality. However after it passes a certain threshold human ears cannot tell the difference. If a sampling rate which is high compared to the frequency of the signal is used, the outcome signal is of good quality since there are several samples to represent each period of the signal. If an increasingly smaller sampling rate is used on the same signal then at one point there will not be enough samples representing each period to give the correct representation of the signal's frequency. An example can be seen in Figure **??**Sampling) showing how a smaller sampling rate than the signal's frequency is used and creates a signal with lower frequency than the original.

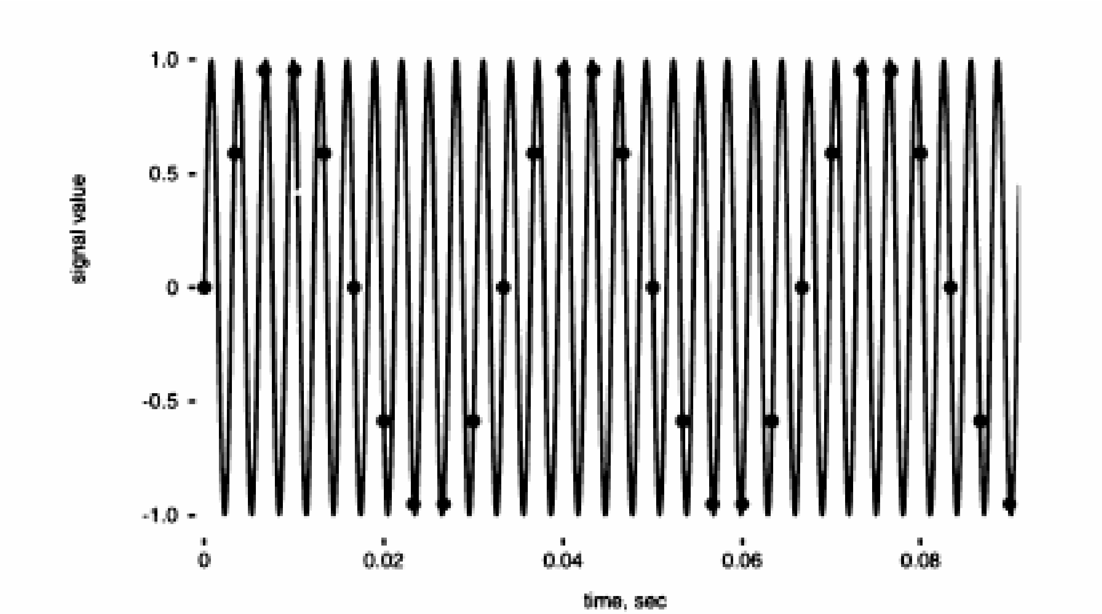

Figure 4.1: Sampling a 330 Hz signal at the rate of 300 Hz[**?**]

The most common sampling rate used when it is needed to capture the entire range of human hearing(20-20000 Hz) are 44.1 kHz(same as used in compact discs), 48 kHz, 88, and 96 kHz. The reason for a sample rate to be twice as high as the signal's frequency is due to the Nyquist theorem which claims that a double rate is enough to perfectly reconstruct the signal. While it is possible to reconstruct a signal without the double sampling rate, other constraints of the signal need to be known. For the prototype a sampling rate of 48 kHz was chosen. A higher sampling rate is not necessary as a higher sampled signal than 50 kHz does not provide any additional information to a human listener. A lower sampling frequency as 44.1 kHz could be used with the same effects, but one lower than that should not be used.

4.0.2 Pitch Shift

The pitch shifter is built based on the principle of a rotating tape head, instead of using six tape heads as a machine would, the code uses two[3]. The tape heads read the signal at a speed independent from the tape speed, thus the speed of the tape determines the duration of the sound and the direction of tape heads control the pitch shift. When the head rotates in the same direction as the tape, the pitch is lowered and if it rotates in the other direction, the pitch is increased, as seen in figure 4.2.

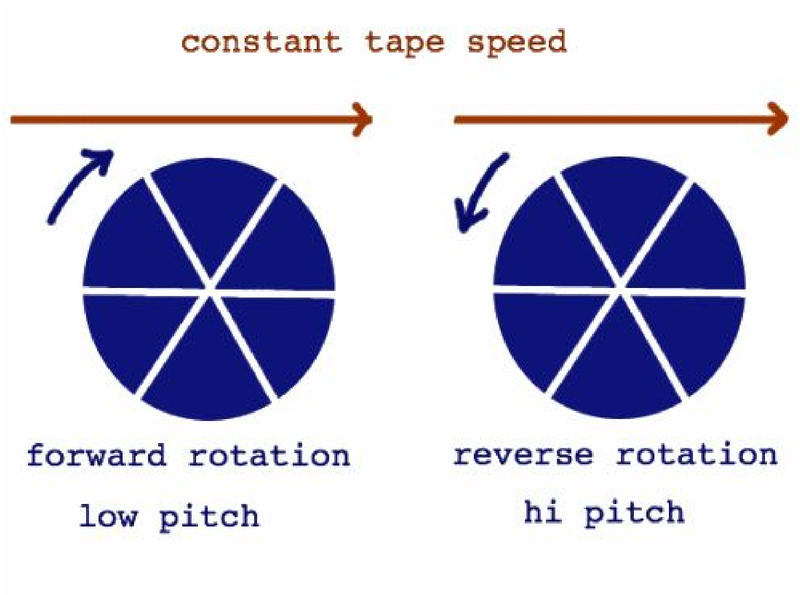

Figure 4.2: Graphical Impression of the tape head principle[3]

In a digital setting the tape heads are replaced with two read pointers that change their delay depending on the required pitch shift. Increasing the delay when lowered pitch is needed and decreasing it if a raised pitch is needed. The read pointer will at some point pass the write pointer, because of the delay. When this happens a click can be heard if audio from only one read point is played. To combat this the cos object is used which creates a smooth crossfade to the second read pointer.

4.0.3 Harmonise

A harmony is a sound created by playing or singing different notes at the same time[15]. Singing harmony means that a backup vocalist or someone else is singing the needed notes in conjunction with the lead singer singing the main melody notes. In a digital setting this effect can be achieved with a single voice. This is done by pitch-shifting the voice multiple times, each time with a different shift in semitones and stacking them together creating the effect of singing in harmony. A chord or a harmony is multiple notes, usually three or more, played at the same time. A chord in which three notes are played is called a "triad"[16]. A triad is build on thirds, a root, a third, and a fifth note, see figure 4.3. The third being a third above the root and fifth being a third above third, or a fifth above the root. Depending on the quality of the two third intervals the quality of the triad changes.

Figure 4.3: A major, minor, diminished, and a augmented triad [17]

In western music there are seven kinds of harmonies, three major, three minor and one diminished. The major triad and the minor triad will be implemented in the program of this project.

5. Implementation

This chapter describes the theory behind the technical aspects of the prototype and how they were implemented into the prototype. This includes the necessary hardware to make the prototype and how these were used in conjunction with the software to make the prototype work. The software includes both the operations of the hardware and the musical processing of the input sounds.

5.1 Making the Prototype

This section describes the hardware for the system and how it was made into a working prototype. This includes both the schematic, which explains how the system is put together, and the software for the prototype which both covers the software for the Arduino and sensor, and how they work together with Pure Data to process the input sounds for both harmonizing and pitch shift.

5.1.1 Hardware

The prototype consists of an Arduino Mega 2560[18], with a connection to a circuit board with an inertial measurement unit(IMU). The IMU uses the MPU 9150 sensor with 9 degrees of freedom[19]. It has a tri-axis gyroscope, magnetometer, and accelerometer.

Write about Gyroscope and why we use fingers

Schematic

The following figure 5.1 shows the connection from the Arduino to the circuit board with the IMU. The black switch at the top of the circuit board represents the three connections, which is put on the thumb, index and middle finger. These are velcro based, with copper tape on them, to create the connection when pressed together.

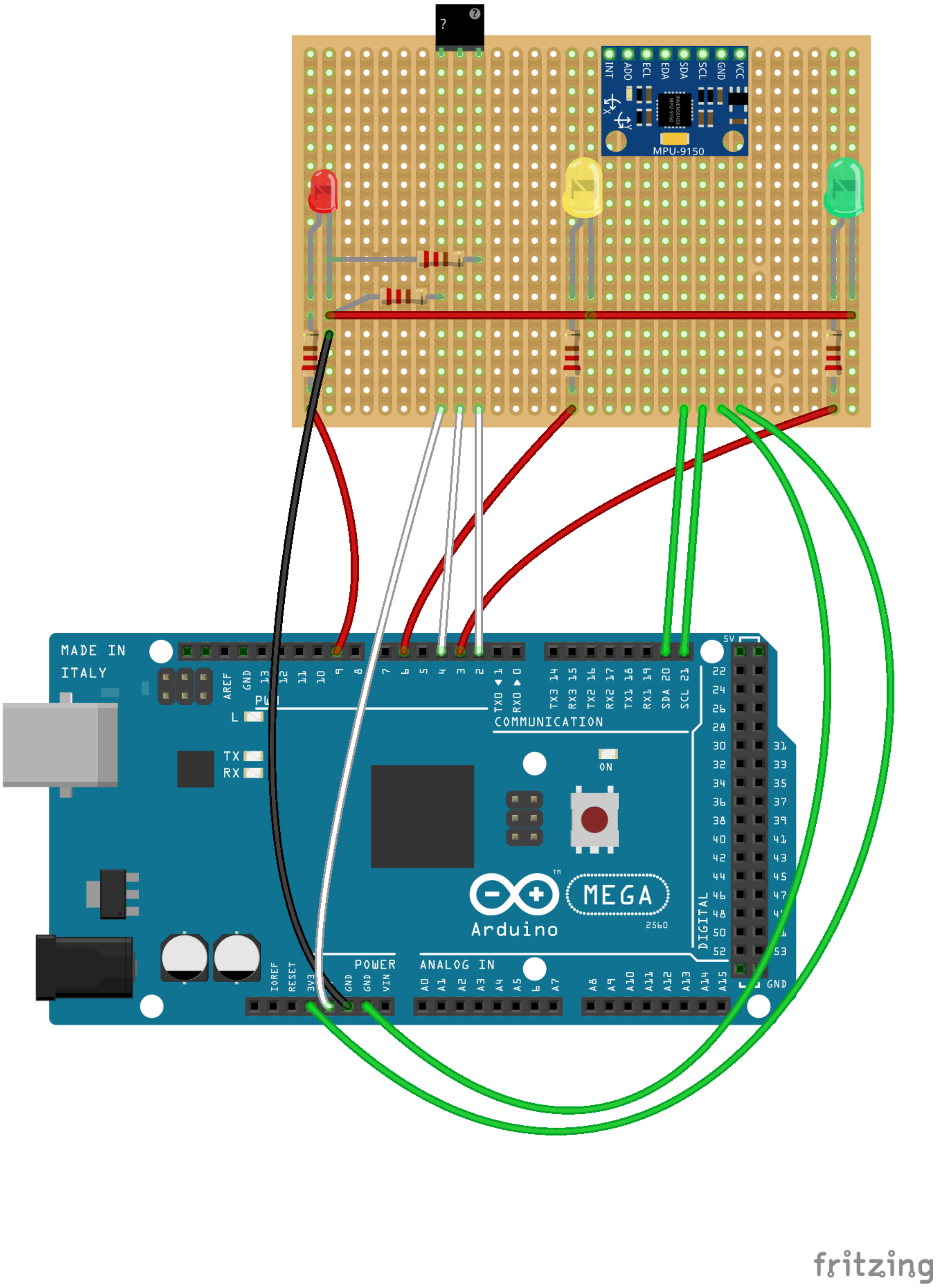

Figure 5.1: Prototype Schematic

The circuit board consists of three LEDs, a yellow, green and red. They light up at different points, which is explained in the next section of this chapter. The LEDs are connected to resistors of 220 Ohm, which are wired to the Arduino as seen in the schematic. The green LED is connected to pin 3. The yellow LED is connected to pin 6 and the red is connected to pin 9. This is essential to make it work in the

Arduino code, which will be described later. The red wire connects all the LEDs and the black wire runs to the ground(GND) on the Arduino. The little blue circuit board on the yellow board represents the IMU. The necessary connections are: SDA, SCL, GND and VCC. SDA is the data line that sends data between the two devices, and the SCL is the clock line that sends pulses at a regular interval[20]. Every time the SCL changes from low to high, a single bit of information is send over the SDA. The SDA connection is connected to pin 20 and the SCL is connected to pin 21. GND is connected to ground on the Arduino and the VCC connection powers the sensor and is therefore connected to 3.3V on the Arduino. The white wires are also connected to the Arduino. The first wire from the left, which goes to the thumb, is connected to the 5V on the Arduino. The white wire beside it, is connected to the index finger, and pin four on the Arduino. The last white wire is connected to the middle finger and pin two on the Arduino. The wire which is connected to the thumb creates a circuit to the LEDs whenever the connection is created between either the thumb and index finger or thumb and middle finger.

The LEDs has to be connected to a pull-down resistor[21], which makes sure, that the Arduino does not 'float' between two different values. A pull-down resistor ensures, that the value is zero, when no active device is connected. The following figure 5.2 shows how a circuit with a pull-down resistor might look.

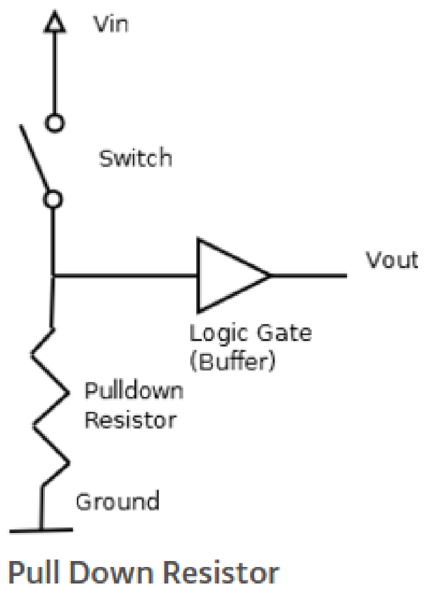

Figure 5.2: Schematic of Pull-down resistor[21]

Figure **??** shows the prototype. As one can see, the velcro for the fingers has copper tape on it, which has been glued onto the velcro. The velcro enables the user to wear the glove, and use it properly. This means that whenever the copper tape gets connected with the copper tape on the thumb, it creates the connection as mentioned earlier. The prototype has cloth sewed onto it, which makes it possible to wrap it around the wrist.

Picture with front and back of prototype here

5.1.2 Software

The following section describes the software part of the implementation.

Arduino

The Arduino language is based on C/C++, and whenever a sketch is compiled, it is sent to a C/C++ compiler [22].

Information from the IMU is continually read and send to the Arduino program. This data is then calculated by the Arduino program. It gives the pose, the angle, and the current movement of the IMU. All based on the data from the gyroscope, accelerometer and magnetometer.

The first part of the code shown in figure 5.3 reads the state of the copper plates. The following if-statement is executed whenever the index finger is connected with the thumb. If the user turns the hand to the left, the code sends the information to PureData, and if the hand is turned to the right, it sends that information. The direction of the hand determines the pitch within a value of 0-255. The motion is a rolling motion, which resembles the motion of turning a knob.

```
//Reading copper plates/buttons
isOn2 = digitalRead(2);
isOn4 = digitalRead(4);


//Index Finger
if (isOn2 == HIGH) {

    if(toggle==false)
    {
     Serial.write(2);
     toggle=true;
    }

  if (pitch > 255) {
    digitalWrite(9, HIGH);
    Serial.write(255);
  } else if (pitch < -255) {
    digitalWrite(3, HIGH);
    Serial.write(-255);

  } else if (pitch < 255 && pitch > 0) {
    analogWrite(9, abs(pitch) * 3);
    digitalWrite(3, LOW);
    Serial.write(map(ardVal,0,255,10,255));
  } else if (pitch > -255 && pitch < 0) {
    analogWrite(3, abs(pitch));
    digitalWrite(9, LOW);
    Serial.write(map(ardVal,0,-255,-10,-255));
  }
```

Figure 5.3: Defining what happens when the index finger is connected

The code used for the middle finger is almost identical. It does however use different pins. Initially it was the plan to use the raise/lower movement for the pitch shifting, but since it caused some errors, it was decided to use the rolling motion for pitch also.

Figure 5.4 shows the last part of the code. This is an else statement that makes sure that the button is off whenever the copper plate is released.

```
} else {
  digitalWrite(9, LOW);
  digitalWrite(6, LOW);
  digitalWrite(3, LOW);

  //Copper button is "pressed" once again.
  //This makes sure the button is off when the copper plate is released
  if(toggle==true){
   Serial.write(2);
   toggle=false;
  }else if(toggle2 == true){
    Serial.write(4);
    toggle2=false;
  }
  Serial.write(0);
```

Figure 5.4: Else statement which makes sure the button is off when released

Pure Data

The audio processing has been done in Pure Data(PD)[23], which is an open source programming language. It is used to generate and process sound, in a graphical way. The program uses patches where one can create objects which makes it possible to create different audio effects. The following subsection describes how PD has been used in this project, and explains the patches made for this project.

The first figure shows the main patch of the PD program, see figure 5.5. The figure shows three coloured rectangles, red, green, and blue. The content of the red rectangle is Arduino related. PD gets data from the Arduino from the "comport" object, which is a serial port interface. The comport objects needs a device number, and a baudrate, in our case device 1, and the baudrate 4800 (must be the same as the Arduino program baudrate). The comport object takes three message objects as input, the name of USB-device which starts the comport object, a message which shows which ports are available, and a close message.

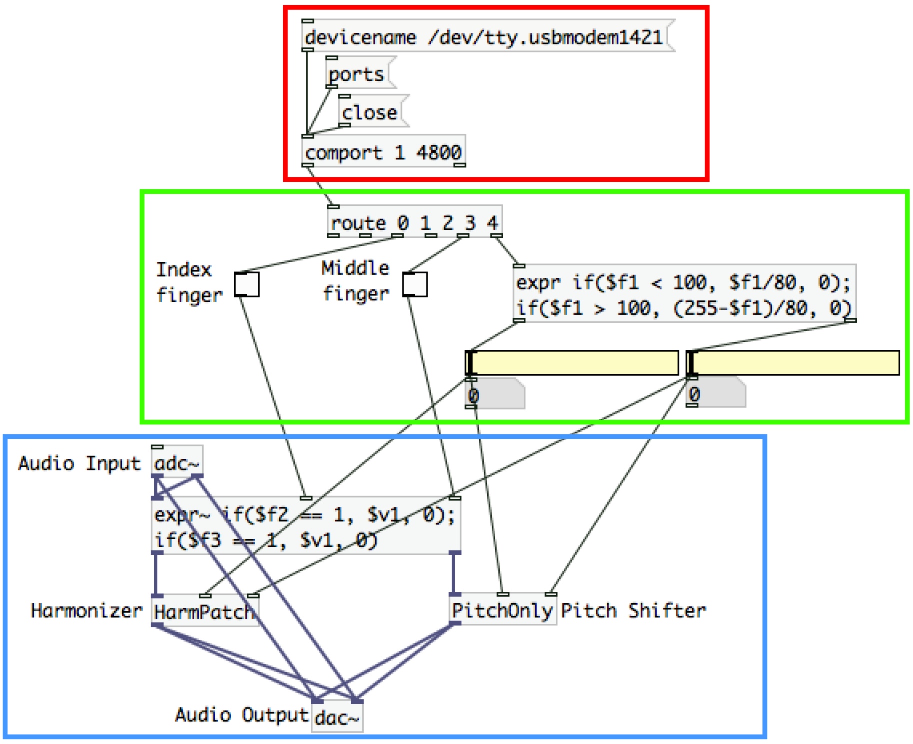

Figure 5.5: The PD Patch. Red rectangle is the Arduino part, green is variable part, and blue is effects part.

The green rectangle processes the output from the comport object. The "route" object takes the data from the comport and splits it up into five parts. Route output number two and four correspond to the copper-plate buttons. E.g. if the index finger is connected, route output number two activates the toggle. The last route output is the sensor data, which goes to an "expr" object. The positive sensor data is between 10 and 100, and the negative between 255 and 155. The "expr" object splits the positive and "negative" values from each other, and divides them with 80 (the "negative" values are subtracted to 255 first). The values go to two sliders.

The blue rectangle does the audio processing. Firstly, the "adc~" object (analog to digital) takes the microphone input and goes into a "expr~" object. If the index finger is connected, the audio goes to the "HarmPatch" patch, and if the middle finger is connected, the audio goes to the "PitchOnly" patch. Both the "HarmPatch" and "PitchOnly" takes the arguments of the sliders from the green rectangle. The outputs from the effects then goes to the "dac~" object (digital to analog), which will go to the headphones.

See figure 5.6 for the "HarmPatch" patch used for harmonising. As you can see in the figure, the audio coming from the "inlet~" goes to four "PitchShifters". The

"PitchShifter" object takes a message as an argument, which corresponds to number of halftones one would like to pitch-shift. The two first "Pitchshifters" make a minor chord by pitchshifting by three and seven halftones. The last two "PitchShifters" make a major chord by pitchshifting by four and seven halftones. The outputs from the four "PitchShifters" are then added in an "expr~" object, which takes the sensor input from the green rectangle from figure 5.5. The output from the "expr~" object then goes to the "outlet" object.

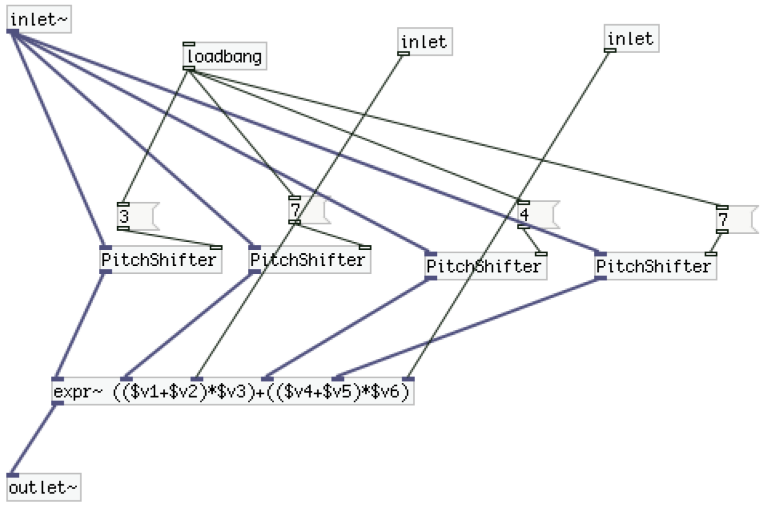

Figure 5.6: The "HarmPatch" Patch. Audio goes to four PitchShift patches, and are added again after

See figure 5.7 for the Pitchshifting patch used in the harmoniser effect. The green rectangle receives a message, the halftone number, but must first convert it to a value that PD will be used later. The conversion uses this equation (REF):

$$2^{h/12} = e^{0.05776*h}$$

where h is the halftone number. E.g. three halftones would give the converted number 1.189.

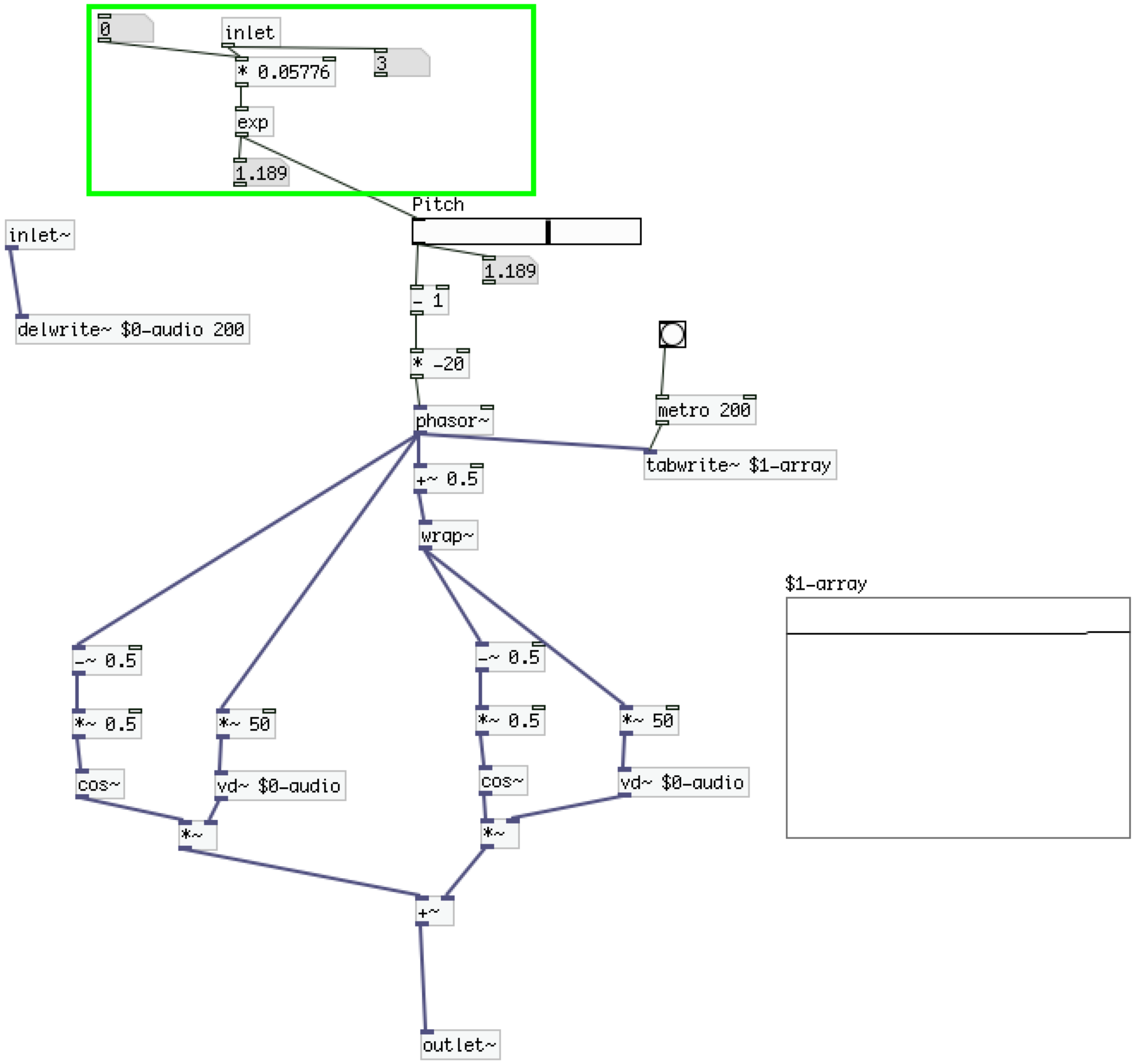

Figure 5.7: The Pitchshifting Patch used for the harmonize effect. The green rectangle is a bit different compared to the Pitchshifting effect.

The pitchshifting effect's patch does not convert numbers, but instead gets the slider data from the main patch, see figure 5.8. Since the pitchshifting effect takes the input from 0-2, where 1 is regular pitch, the slider data is either added or subtracted with 1.

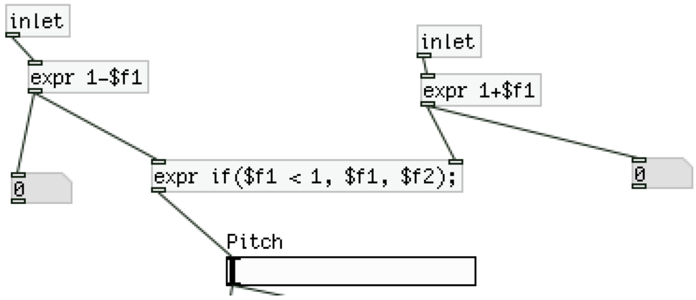

Figure 5.8: How the Pitchshifting patch handles slider data

The red box in figure 5.9 shows the conversion from halftone to pitch-shift value, as discussed earlier.

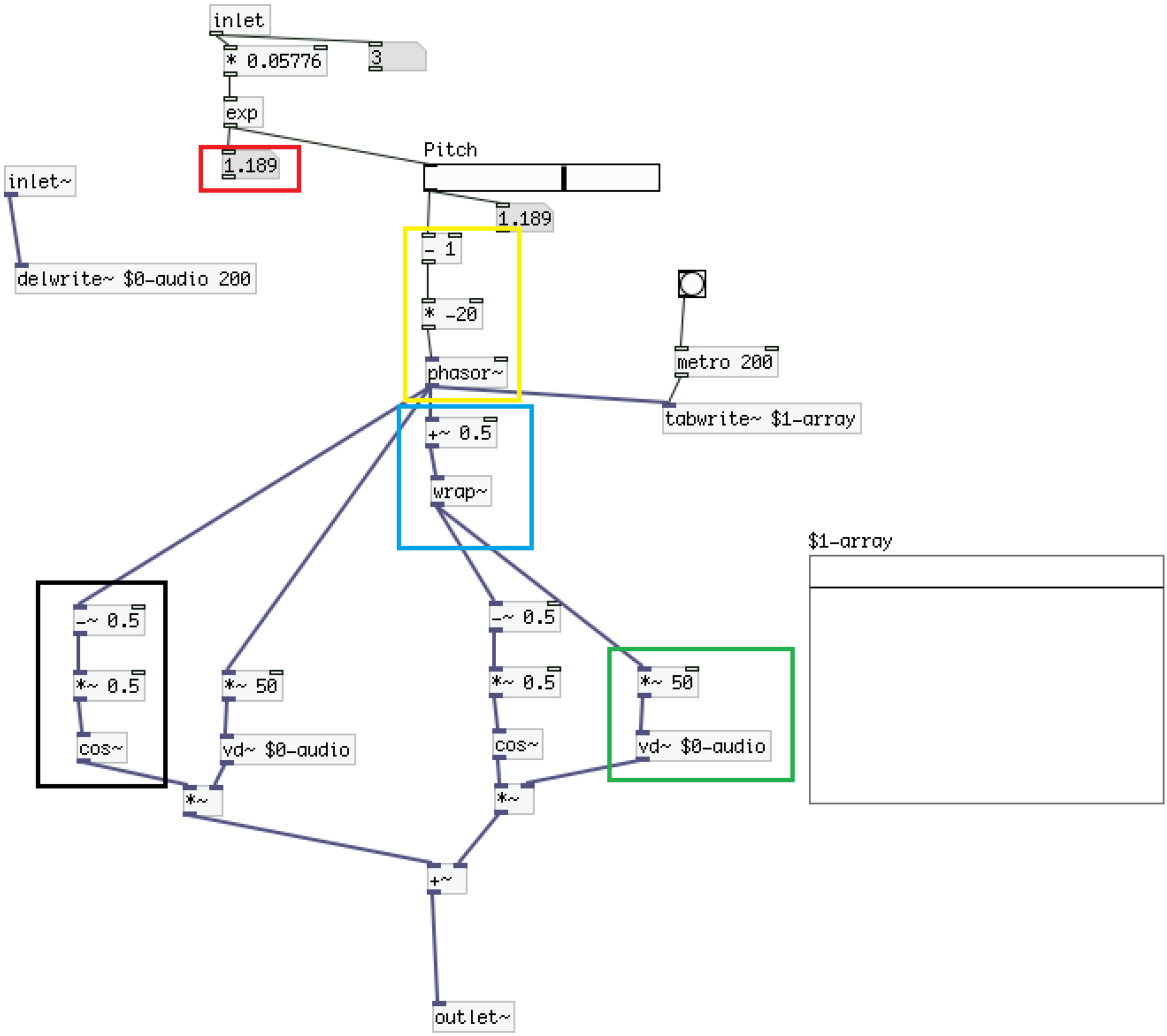

Figure 5.9: The pitch patch

The "tapes" speed is 1 so to make the pitch shift, one of the read points scans the sound, delayed but at a faster rate than the "tape" is playing, so that the delay progressively goes to 0. To support this the sawtooth signal is inverted allowing for the delay to start at maximum values and to go to minimal values before resetting back to the maximum values. This is shown in the yellow box on figure 5.9. The green box on figure 5.9 illustrates the read point objects, which each scan 20 frames of the signal per second, which means that each frame is 50 ms long. The blue box on figure 5.9 shows the offset between read points created by the + 0.5, so that when one of them is at the maximal or minimal value the other one is in the middle. Finally, the black box on figure 5.9 describes how the patch removes glitch objects. When one of the read pointers would reach maximum delay or 0, there would be an audible glitch if audio from only one of them is taken. To remove the glitch objects cos are used for the smooth crossfade between both read points.

5.2 Conclusion

6. Evaluation

The following chapter describes the evaluation and introduces the results from the testing of the prototype.

There were conducted two tests, an internal and a user test. This was done to test whether the prototype was functional on a technical level and to test whether is was usable by people not part of this project.

6.1 Internal Test

To test the accuracy of the prototype, an internal test was carried out. The following hypotheses have been stated for the internal test:

H_0: The prototype works less than 90% of the time.

H_A: The prototype works more than or equal to 90% of the time.

The test consisted of the copper plates being connected 25 times each. First the index finger were tested with the thumb afterwards the middle finger with the thumb. Whenever the yellow light on the system lit up it was considered a success. If the light did not turn on it was considered a failure. The times it succeeded and failed were counted.

The results ended up being as follows:

- Index finger connection test: 23 out of 25.
- Middle finger connection test: 25 out of 25.

The results have been analysed statistically using R[24]. R is a language used for statistical analysis. All the statistics have been calculated with a probability of success, p, of 0.5.
The results from the index finger connection test showed a p-value of:

$$1.943 * 10^{-5} \tag{6.1}$$

Whenever the p-value is less than the 0.05 significance level, a conclusion can be drawn from the test. In this case the H_0 is rejected for the index finger, since the test gave a probability of success of 0.92(92%). prototype worked at least 90% of the time when connecting the index finger and thumb.

The results for the middle finger connection test showed a p-value of:

$$5.96 * 10^{-8} \tag{6.2}$$

The p-value lies below the 0.05 significance level, so a conclusion can be made. This means that this connection works at least 90% of the time, and the H_0 is rejected for the middle finger also.

Another internal test was carried out to test the accuracy of the effects. The test was conducted in a similar way, the effects were applied and then used. As long as the effect worked and changed the voice input, it was considered a success.

In the first test, the system worked 18 out of 25 times for both gestures. Since the gesture effects showed the same results, the p-value is the same for both gestures. The p-value ended up being 0.04329 which is lower than the 0.05 significance level and therefore the result can be concluded upon. Thus, the H_0 is accepted, since the prototype worked less than 90% of the time with the gestures.

During the first tests, the system had a loose wire. The issue was fixed and the tests were then conducted once again with much better results:
All the tests worked 25 out of 25 times.

This gave a p-value equal to equation 6.2, which is much less than the 0.05 significance level and a probability of success equal to 1(100%) and therefore H_0 are rejected for all the tests.

6.2 User Test

When conducting this user test, it was the intention to get some data from the users about the different effects and gestures.

6.2.1 Evaluation Plan

To evaluate the working prototype three hypotheses based on the device were stated. They are as follows:

1. Participants understand the link between gesture and effect
2. The design is awkward to wear

3. The participants will have no clear preference of effect

To evaluate the hypotheses a test was conducted with the goal to either prove or disprove them. First the participants were asked to sign a consent form which together with the script can be found here: Appendix **??** and **??**. The test then consisted of the participants doing different tasks, which was followed by a few questions to determine whether to prove or disprove the hypotheses. The results of those questions can be found in **??** The questions can be found her Appendix **??**.

6.2.2 Apparatus and Setup

The following equipment was used to perform the test:

- Two laptops
- Zoom H4N microphone
- Headphones
- The Prototype

The following figure 6.1 shows the setup of the evaluation. One person wearing the system, a facilitator, and one person that makes sure the system works and has the microphone in his left hand (technician).

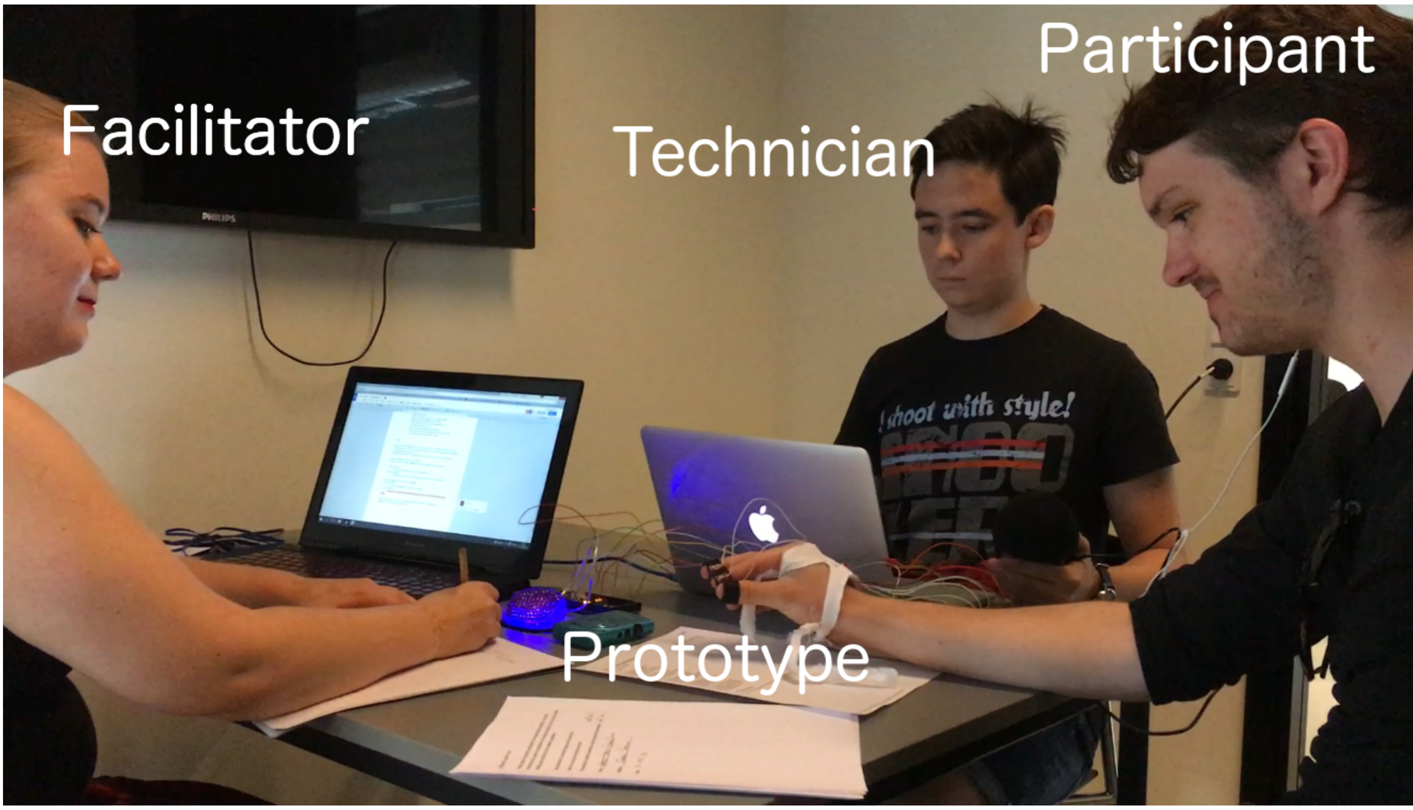

Figure 6.1: The Test Setup. There is a facilitator, a technician, and a participants. The technician has the microphone in his left hand.

6.2.3 Results

The hypotheses previously stated will be rejected or kept based on the results from the questions asked during the interview.

The first hypothesis deals with the link between the gesture and the effect. Here the question were asked: Did the gesture makes sense together with the effect? All the participants answered yes to this and also said why it was. A female participant said "[it is] like turning a button up and down". A male participant mentioned that "It is like turning [the volume] up and down". Another male participant said "[it is] like turning a knob". These three are examples of the exact idea the gesture was based upon.

The second hypothesis deals with the physical aspects of the prototype. Here the question asked was: Was the device comfortable to wear? Again all participants agreed. The device was comfortable to wear.

The third hypothesis deals with preference of effect. The participants were asked: Which effect did you prefer? All participants, except one, answered pitch-shifting.

This presents a clear preference for the pitch-shifting effect.

Aside from what the quantitative data from the questions, there were some additional data. Mainly the comments and feedback from the participants during the test which is analysed as qualitative data. Especially the participant with a musical background gave a lot of useful feedback regarding the prototype. Four out of the ten participants stated that they could not hear a difference between the major and minor harmonics. The participant with a musical background heard the difference clearly. However an assumption could then be made that the other participants could not hear the difference since they did not know what they were listening for.

They could all clearly hear the difference between shifting the pitch up or down. However five participants mentioned that the pitch-shifting is very sensitive and that there is not a lot of middle ground between no effect and full effect.

Another thing mentioned by three of the participants is that even though all participants found the gestures intuitive, holding them for longer periods of time could be very straining and cramp-inducing on their hand.

When asked whether they thought the prototype useful or not, nine of the ten participants said that they thought it could be. Some said that it could be useful in theatre performance and another said that it gives more control to the singer. Both of these are some of the ideas this project is based on.

The one participant we had who had a musical background also gave some additional constructive feedback. She stated that the harmonise effect lacks the ability to be able to choose the intervals you want. She said that it would be "incredibly smart if the first notes added the 3rd interval and then the 5th interval, and maybe even removed the 3rd interval if you twisted the wrist even further. Then you are in more control of the notes and the expression the music makes." Changing the intervals is something to consider for further iterations.

6.3 Discussion

Several parameters of these tests must be considered before any conclusions are drawn. Only one participant from this user test fits this project's target group. This does affect the results quite a bit since participants of this test do not have the necessary knowledge to know how the prototype could be utilised, and therefore cannot give very useful feedback on the musical aspect of the prototype. Then there is the issue that the first half of the participants tested a prototype with a loose wire which made it act oddly and made it more prone to not react to the gestures made by the participant. The questions were also mostly yes/no questions which gives a rather limited view of the participants' reactions and views of the prototype. A

Likert scale would maybe have been more appropriate for more nuanced answers.

6.4 Conclusion

Based on the results from these two tests it can be concluded that the technical parts of the prototype works as intended. Even though the system is simple, it works every time and all the participants like the concept and thought it useful, even though they may not have correct knowledge to deem it so. Using the feedback that was gathered, certain adjustments could be implemented, ie. less sensitivity to the pitch-shifting and more adjustability to the harmoniser. The concept of the prototype working with the hands works very well although further exploration of different gestures and effects could be a good idea.

7. Discussion

The problems this project address is that singers or performers cannot apply voice effects easily. The problem analysis chapter covered which kind of voice effects exist today, current voice effect pedals and interfaces, studies regarding voice effects and gestures, and what a gesture is and how it can be used.

There are two kinds of effect interfaces, some are on the floor and highly customisable but hard to reach/change while performing. Some effect interfaces are small and on a microphone stand, but are hard to customise. We had no qualitative or quantitative data telling us if there was a problem.

During state of the art research, it was found that there are studies focusing on using gestures to change effects, and a project that focuses on a highly advanced system. The studies focusing on gestures that change effects, “HandySinger” and “One Person Choir” found that having gestures which make sense to the user is both possible and beneficial. Additionally, a project with a very similar concept was discovered, the Mi.Mu glove. Their project was focused around making a glove for musical interaction with a wide variety of features. In order to differ from their project, we decided to focus our project around making a glove with less features, but much more focus on making it intuitive for the user.

Many voice effects exist. Instead of turning an effect on and off, changing an effect parameter could sound better or make more sense. We chose the effects “harmonise” and “pitch shift” because they could benefit from continuous change, and because a parameter change would be easier to hear compared to an effect like the reverb. This does not mean effects like reverb, delay, vocoder etc. are bad effects to change parameters to, but rather effects that could be added in further development.

When deciding on a fitting gesture for the different effects introduced during the problem analysis, it was important to think about each effect individually. Should they just turn on and off or are there parameters that needs to be changed? Which parameters needs to be changed and in what way? In the case of our project, it was important to be able to change the parameters of each effect. The pitch shift needs to be able to lower and heighten based on the user’s wishes. The harmonise should be able to change between minor and major, and the degree of which it can be heard. It was decided the most fitting gesture for pitch shift was to simulate a vertical slider, moving to heighten and down to lower. The gesture for harmonise was to simulate a knob, like when changing the volume on an old radio.

The prototype was designed to be worn on the right hand on the thumb, index and middle fingers. At this point, the prototype must be connected to an Arduino with a lot of wires, which means the user has limited movability. We also found that there was no room for effect icons even though we had drawn them in the Lo-Fi prototype. This will be considered in further development.

The prototype only understands the knob-turning gesture. We did manage to get the horizontal slider gesture working, but it was not stable enough to include in the evaluation tests. At first, the sampling rate of the Pure Data program was at 44.100 Hz, but it was found that it created unwanted audio problems sometimes, such as a delay, and was fixed by using a sampling rate of 48.000 Hz.

7.1 Quality of Solution

This section will discuss the qualities of the system. Our first success criteria states that we have to implement at least two effects.The prototype can process audio, more specifically take the microphone input and apply two effects, whose parameters users can change. The two chosen effects ended up being a harmoniser and a pitch-shifter, because they are easy to hear, and would benefit from gesture manipulation. A reverb effects was also considered but later discarded, since we felt that the effect did not fully benefit as much from parameter change in a live performance.

The second success criteria stated that the user should use the correct gesture for the intended effect. This was based off of the fact that we wanted to implement different gestures at the time. During the implementation it was discovered that the slider gesture worked, but was not reliable. So we decided to use the knob-turning gesture for both effects. Given this, the third success criteria, stating that the system should not misinterpret between gestures, was not fully achieved. Beyond that, the system had no problem recognising the knob-turning gesture.

The system fulfils all of the minimum implementation requirements by using an Arduino, utilising a sensor that works with the Arduino, implementing the system using audio processing in PD and connecting it with the Arduino software, and using a microphone to record the audio.

Even though the success criteria are approximately fulfilled there were some issues with the prototype. The prototype has many long wires attached to the Arduino, which is attached to a PC, and can accidentally disconnect from the Arduino. This limited the movements of the participants of the test, which might have affected the results of the test.

A problem that was encountered was that some users using the prototype for a long period of time, started to feel discomfort in their palms. Additionally, the users missed a source of feedback when turning their hand over too much, causing the LEDs to disappear from view. This might have affected their understanding of what was going on. Another thing to consider is the plus and minus icons that were designed to indicate what was happening with the parameters when tilting the hand. Plus and minus is not necessarily the best signs for both of the effects, but it was decided to go with this since it was hard to find a uniform indication that fit both effects.This may not make sense to some of the users.

7.2 Validity and Reliability

A problem we acknowledge is the fact that nine out of ten users that tested our system were not singers or performers. They had basic knowledge about some effects, but no experience performing. This means we cannot consider their answers, about whether it would be beneficial for a performer, valid, but rather look at their answers from a usability standpoint. On a note, the nine out of ten participants were Medialogy students, which means they may be a little biased regarding the topic of the project. We had one participant, a non-Medialogy student, who is within our target group and therefore can be considered valid. However one participant cannot be considered reliable, since reliability requires replication.

Regarding the actual prototype, the soldering on the circuit board and copper plates on the fingers was not optimal, and had to be fixed a couple of times. This resulted in some users trying a prototype that was faulty which also might have affected the results of the test.

The test was conducted in a small room and the user was using headphones to listen to the output of the system. The ideal setting would be on a stage and with a normal performance setup - this would be a dynamic microphone and speakers so as to avoid a feedback loop. Furthermore, the user was the only one hearing his or her voice with effects, meaning no one conducting the test could hear the output. By watching the PD program and LEDs it could be observed whether or not it was working properly.

7.3 Further Development

During evaluation several ideas arose on how to improve the prototype. Some of them were obvious, for example making the prototype wireless. This would make it easier to manage and manoeuvre. Another rather obvious improvement would be to add more effects on the remaining fingers. This could be done by using different kind of sensors, e.g. flex sensors on the fingers. Which effect these are to be are not clear but the possibility is there. More minute control of the effects would also be an improvement. This would mean that there would be a more gradual change to changing the pitch or more control of which harmonies are added by the harmoniser. An improvement of the design would also be on the table. A more easily wearable design, that one person could put on alone, would be a lot more useful and comfortable. A design similar to the Mi.Mu glove introduced in the State of the Art could be considered.. It would also be interesting to see if the colour of the LEDs would matter or not.

8. Conclusion

Bibliography

[1] C. Loeffler, “A brief history of time-based effects (part 1).” http://www.harmonycentral.com/articles/exploring-time-based-effects, September 2014.

[2] N. Redmon, “A bit about reverb.” http://www.earlevel.com/main/1997/01/19/a-bit-about-reverb/, January 1997.

[3] Katjaas.nl, “Pitch shifting.” http://www.katjaas.nl/pitchshift/pitchshift.html.

[4] M. Dolson, “The phase vocoder: A tutorial,” *Computer Music Journal*, vol. 10, no. 4, pp. 14–27, 1986.

[5] A. Hadhazy, “What’s auto-tune and how does it work?.” http://www.livescience.com/11046-auto-tune-work.html, September 2010.

[6] Innovativesynthesis.com, “Introduction to vocoders.” http://www.innovativesynthesis.com/introduction-to-vocoders/.

[7] TC-Helicon.com, “Perform v.” http://www.tc-helicon.com/en/products/perform-v/.

[8] EHX.com, “Vocal harmony machine/vocoder.” http://www.ehx.com/products/voice-box.

[9] I. Heap, T. Mitchell, K. Snook, S. Madgwick, H. Perner-Wilson, A. Stark, R. Freire, and C. V. D. Berg, “Mi.mu gloves.” http://mimugloves.com/, 2015.

[10] T. Yonezawa, N. Suzuki, K. Mase, and K. Kogure, “Handysinger: Expressive singing voice morphing using personified hand-puppet interface.” http://www.nime.org/proceedings/2005/nime2005_121.pdf, 2005.

[11] P.-J. Maes, M. Leman, K. Kochman, M. Lesaffre, and M. Demey, “The one-person choir: A multidisciplinary approach to the development of an embodied human–computer interface.” http://www.mitpressjournals.org/doi/pdf/10.1162/COMJ_a_00054, 2011.

[12] D. Wigdor and D. Wixon, "Brave nui world." http://www.gm.fh-koeln.de/%7ehk/lehre/sgmci/ss2015/Literatur/Wigdor_Wixon_-_Brave_NUI_World.pdf, 2011.

[13] V. Kaptelinin, "Affordances." https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/affordances.

[14] AudioFanzine, "Ax 301 synthesizer amplifier." http://en.audiofanzine.com/keyboard-amplifier/moog-music/ax-301-synthesizer-amplifier/medias/.

[15] "Music theory: Harmony part one (basics)." https://www.mymusicmasterclass.com/blog/music-theory-and-composition-basics-pt-1/.

[16] MusicAwareness.com, "7 harmonies tutorial." http://www.musicawareness.com/Tutorial3.html.

[17] Johnston, "Triads – major and minor." http://blogs.yis.ac.jp/johnstonb/theory-2/triads-major-and-minor/.

[18] "Arduino mega 2560." http://www.arduino.cc/en/Main/ArduinoBoardMega2560.

[19] Sparfun, "Sparkfun 9 degrees of freedom." https://www.sparkfun.com/products/retired/11486.

[20] Arduino.cc, "Master writer/slave receiver." https://www.arduino.cc/en/Tutorial/MasterWriter, August 2015.

[21] "Pull down resistor." http://playground.arduino.cc/CommonTopics/PullUpDownResistor.

[22] Arduino.cc, "Frequently asked questions." http://www.arduino.cc/en/main/faq.

[23] PureData.info, "Pure data." https://puredata.info/.

[24] T. R. Foundation, "Introduction to r." https://www.r-project.org/about.html.

A. Appendix

A.X - Results from Questionnaire

A.1 Results from Questionnaire

Gender	Age	Do you sing?	Have you ever performed with voice effects applied?
Female	18-25	Yes	No
Male	18-25	No	No
Male	18-25	No	No
Female	18-25	Yes	No
Male	18-25	No	No
Female	18-25	Yes	No
Female	18-25	No	No
Male	18-25	No	No
Male	18-25	No	No
Male	18-25	No	No

Was the device comfortable to wear?	Did the effects work as they were supposed to?
Yes	Yes
Yes	Yes
Yes	Yes
Yes	No
Yes	Yes
Yes	Yes
Yes	Yes
Yes	Yes
Yes	Yes
Yes	Yes

Did you like the effects?	Which effect did you prefer?
Yes	Pitch-shifting
Yes	Pitch-shifting
Yes	Harmonise
Yes	Pitch-shifting
Yes	Pitch-shifting
Yes	Pitch-shifting
Yes	Pitch-shifting
Yes	Pitch-shifting
Yes	Pitch-shifting
Yes	Pitch-shifting

Did the gesture make sense together with the effect?	Did you like the concept of the device?
Yes	Yes
Yes	Yes
Yes	Yes
Yes	Yes
Yes	Yes
Yes	Yes
Yes	Yes
Yes	Yes
Yes	Yes
Yes	Yes

Do you think it would be useful to performers?	Why?
Yes	If they can wear it on stage. Maybe in theatre.
Yes	
Yes	It could be cool
Yes	The harmonise would be useful. I see no use for the pitch for performers.
Yes	It is easy to apply effects.
Yes	It gives more control to the performer.
Yes	It could be useful in some cases such as theatre.
Yes	Maybe it should be moved from the hand to the microphone. Other solution needed than pressing the fingers together.
No	There already exists so much else in this category.
Yes	One just needs to adjust and then it would be useful.