BAES - a Bass Augmentation and Enhancement System

- Expanding the Possibilities and Roles for the Double Bass -

Master's thesis - Sound and Music Computing 2017

Mathias L. Damgård & Peter F. Gomez

Aalborg University
School of Information and Communication Technology

AALBORG UNIVERSITY
STUDENT REPORT

Sound and Music Computing
Aalborg University
http://www.aau.dk

Title:
The Base Augmentation and

Theme:
Sound and Music Innovation

Project Period:
Sept. 2016 - June 2017

Project Group:
SMCA171031

Participant(s):
Mathias L. Damgård
Peter F. Gomez

Supervisor(s):
Jesper Rindom Jensen

Copies: 1

Page Numbers: 113

Date of Completion:
May 19, 2017

Abstract:

Through interviews with a student at the Royal Academy of Music in Aalborg, it was discussed that the double bass is traditionally found in jazz, folk and classic where it is part of the rhythm group in a musical constellation. The student, who became our collaborator, wished to change this and challenge the conservative roles through the use of audio effects. Five effects were chosen to give a mixture of common effects such as reverb and delay but they also included more electronically inspired sounds through the combination of the beat repeat effect, convolution and pitch-shifting. The project resulted in the Bass Augmentation and Enhancement System (BAES). A controller was created to house these and in order to fit full parameter control into a relatively small box, motorised faders were used to be able to go back to a previous fader position, thus enabling the use of the same three faders for all five effects. The collaborator used the second iteration of the BAES over a long test period (about a month and a half). During this time, it was used for two different performances. The last interview was made to evaluate the BAES. In this interview, it was made clear that our collaborator could go beyond the traditional role of the double bass and express himself in new ways.

Contents

Preface

As part of the 9th semester of the Sound and Music Computing master's programme, we chose to do a project within the field of Sound and Music Innovation. During the 9th semester, the project was converted into an extended master's thesis giving us more time to go through more iterations and, hopefully, resulting in a close-to-finished product.

In this project report, we describe design choices as well as the actual design and implementation. The report also includes a description of user experience tests as well as discussion and conclusion of the entire project.

The structure of the project is as follows: First the project is introduced in chapter 1 followed by summary of existing SOTA in chapter 2. The report discusses the various considerations needed before making the first prototype as well as how an augmented instrument could be evaluated in chapter 3. Chapter 4 elaborates on the sound effects used, theory and implementation follows. Specific software implementations are covered in 5. The report is then broken down into two iterations found in chapters 6 and 7. Both iterations break down the software and hardware implementation as well as testing. This is followed by a discussion of the project in chapter 8 and the conclusion found in chapter 9. Finally, we discuss possible futures for the project in chapter 10. At the end of the report a paper that was written for the NIME (New Interfaces for Musical Expression) conference is attached (see appendix G). The paper was written during the first iteration and thus does not contain improvements from the second iteration.

Aalborg University, May 19, 2017

Mathias Lyneborg Damgård
<mlda12@student.aau.dk>

Peter Flemming Gomez
<pgomez15@student.aau.dk>

Chapter 1

Introduction

In a musical performance, instrumentalists such as guitarists have a plethora of effect pedals. The pedals are chosen and chained in an order which satisfies the sonic needs of the guitarist. Each individual pedal has a set of knobs and switches to control effect parameters where the parameters of a reverb pedal could be feedback, cut-off frequency for reverberations and damping. Some parameters are even controlled by foot-operated potentiometers (e.g., a Wah-pedal). The effect pedals can then be turned on and off by stepping on a switch with your foot which leaves you free to play your instrument. Changing the knobs with your foot is, on the other hand, highly impractical and makes parameter control all but impossible without having to bend down to the actual pedals.

Our project was done in collaboration with a bass player who plays the double bass - a traditional string instrument found mostly in classical and jazz music. Our collaborator, however, wanted to play his double bass using audio effects as well as being able to change effect parameters on the fly. The double bass is mostly played whilst standing which would make it challenging to manually change parameters real-time on regular guitar pedals (see fig. 1.1). While guitar pedals might be an obvious choice, they are not the best option. Questions such as the following were considered: "What is the inherent role of the double bass?", "What are its capabilities?" and "Does our tool change either of these?".

1.1 Background

Why did we choose to augment the double bass? In a previous project published in the Aalborg University project database, the authors researched convolution as an audio effect and ways to simplify and enable anyone to use this technology in new and novel ways. The project focused on using different signals than room impulse responses for convolution: for instance a plucked cello string or water be poured into a glass. Contact microphones were attached to non-musical objects such as boxes and

Figure 1.1: Double bass player explaining the challenges of moving about when playing the double bass.

water bottles and the signal was then convoluted with the various IRs for interesting audio results. Amongst the test participants was a double bass player. During the test his thoughts went to how he could use convolution with his double bass to create an entirely different experience compared to the traditional way of using the double bass. The bass player, Mads Thorlund Rømer, is, at the time this report was written, a bachelor student at the Royal Danish Academy of Music in Aalborg specialising in the bass guitar which he had played for 15 years and the double bass which he had played for 3 years. After a discussion, we decided to collaborate with him in making a project specifically focused on the double bass and this was the birth of the Bass Augmentation and Enhancement System (BAES). One of the ideas was sparked from our collaborator's impression that solo pieces by double basses could be very diffuse in their sound. The idea of being able to put yourself forward in the soundscape was interesting to our collaborator (see 6.5) but at the same time it was also important to preserve the acoustic sound of the double bass. But before designing and implementing the project, we first reviewed current research on double bass augmentation and augmentation in general.

Chapter 2

State of the Art

In order to gain an understanding of the various products and technologies already on the market, an analysis was made. In this section, we list relevant technologies and summarise at the end of the section.

2.1 Existing Augmentations

One must look at previous augmented instruments to get an idea of what the state of the art is. There are very few (if any) instances of a double bass being augmented, but there are still many things one can learn from any instrument being augmented, as such, this section will specifically look into what has been done before in terms of augmenting an existing instrument.

2.1.1 The Augmentalist

The Augmentalist is an easy-to-use, user-centred system which allows any musician to augment their instruments with ease. The sensors are Phidget sensors which include buttons, FSRs, faders and accelerometers that are easily attached to the instrument of choice [1]. The mapping and interpretation of the sensor data are handled in Max/MSP [2] where it is converted into MIDI signals. The MIDI signals can then be used to control whatever audio software the user desires. For this project one of the most interesting aspects of the Augmentalist project is the close collaboration with musicians and the idea of "musicians as developers"; especially, since they are the experts on their instruments and thus know which augmentations are feasible or not.

2.1.2 Hybrid Violin

An example of a more instrumental specific augmentation is the hybrid violin [3]. The Hybrid Violin concerns their augmented electric violin which they have augmented with sensors and an iPod touch that uses these sensors to control a Pd (PureData) patch that runs on the iPod via MobMuPlat. The important design elements were as follows: high mobility in a battery powered approach; all processing is done directly on the instrument; interfacing is done via the iPod Touch and high accessibility through a low-cost electric violin. The hybrid violin was tested on a string quartet in which they had three violinists and a cellist. The approach was a semi-structured interview, i.e., highly qualitative information. The audio quality and loudness were a problem. Regarding gestures/mapping, test subjects were interested, but they considered the gestures/mappings to be more of a novelty as they lacked exploration, more complex mappings and additional sensings were necessary. It was also important to design the instrument for a specific target group: is it a solo instrument or a group effort? The paper is very focused and precise in both description of the design and evaluation, however, at the time the Hybrid Violin was still very early in its development.

2.1.3 Hybrid Piano

Another example of an instrument specific augmentation is the hybrid piano [4]. Dahlstedt's augmented hybrid piano consists of a piano and a sound processing unit with speaker and microphones placed such that the acoustic and processed sound blend into one - a concept he calls Foldings. The processing techniques include virtual resonance strings, dynamic buffer shuffling and acoustic and virtual feedback. The instrument builds upon the foundation that there should be a correlation between the physical effort exerted playing the instrument and the sound produced. It should be as free and direct as an acoustic instrument with no extra faders or knobs. Evaluation is done via his own qualitative experiences and one other pianist. He argues that this is the necessary approach as one cannot evaluate a complex instrument unless one builds up experience with the instrument over several years. This approach is about trying to understand the possibilities of the instrument rather than saying "this instrument is better than that one" as he puts it. While the paper is interesting it is very far from a double bass, both in terms of sound but also simply the way the instrument is played.

2.1.4 SABRe

The bass clarinet has more in common with the double bass in terms of the frequency content but also the fact that both hands are occupied which is an important aspect when designing an augmented double bass. Therefore, the SABRe [5] was looked into. The authors explore the possibilities of augmenting a bass clarinet to extend the possibilities regarding performance and composition. The project SABRe (Sensor Augmented Bass clarinet Research) includes a plethora of sensors which are processed using a dedicated piece of hardware whereafter the data sent to a computer using OSC. Their goal was to extend the player's possibilities of manipulating effects and signal processing through gestures and sensors without compromising the skills already ac-

Figure 2.1: Control space for the UI.

quired by the player. The last part is an important aspect of this project as the hands of a double bass player are very much occupied when playing; it is important to consider and ensure that the player has easy access to and controls of the desired effects.

2.1.5 The ACPAD

The ACPAD is an invention of Robin Sukroso who wanted to combine elements from electronic music with the sound of his acoustic guitar. The result was the ACPAD which is an easily mountable wireless MIDI controller for the acoustic guitar, which can be seen in fig. 2.1.5. The ACPAD registers user input through piezoelectric disks, buttons, LEDs and capacitive sliders and translates the data into MIDI which can be sent to virtually any software that is MIDI compatible. The piezoelectric disks are used as drum pads which can be used to trigger samples of e.g. drums or sound clips. The buttons are used for starting recording, playback loops, selecting presets and effects as well as enabling effects. The capacitive sliders are used to control different levels e.g. the dry-/wetness of an effect. The ACPAD is interesting for our project as we aim to provide a double bass player with similar controls and options considering effects and their parameters. Another interesting aspect is the ACPAD itself and how it is designed to perfectly fit the acoustic guitar providing easily reachable controls and, thus, extending the capabilities of the guitar. Today the ACPAD is a commercial product that originated from a university project which is very inspiring for us.

2.2 Summary

Many of the design aspects that can be found in the previous examples are useful for our project. An important aspect, that can be found in both the hybrid piano [4] and the Augmentalist [1], is the idea of including the musician in the design process from the start: the tool-maker is also the tool-user as stated by Dahlstedt [4]. Another aspect is the evaluation method which will be explored more in depth in chapter 3, section 3.4. Furthermore, mapping and control are important aspects but as none of the above projects works with double basses, we worked closely with our collaborator to find a feasible set of controls.

Chapter 3

Project Preliminaries

This chapter contains descriptions of how the project started and which design choices and considerations were made before the first prototype was created. It also describes the methods used to evaluate and test throughout the project.

3.1 Problem Formulation

After discussing the roles of the double bass with our collaborator, we proceed with the following problem formulation: "How can we provide tools and possibilities for the double bass to expand its inherent roles and capabilities already found in the instrument?"

3.2 Augmenting the Double Bass

As described in the introduction, one of our challenges was to find a way to give a double bass player access to effects and furthermore the ability to change effect parameters on the fly. Our goal was to augment the double bass in a way that extends its capabilities through the use of sensors, sound production, effects and new modes of interaction.

When augmenting an instrument it is important to consider the player's limitations in terms of bandwidth. The double bass almost constantly requires both hands to play: one hand is used for controlling the pitch through shortening the strings by pressing the string to the fingerboard which also locks the thumb at the back of the neck. The other hand is used to pluck or bow the strings at the area between the end of the fingerboard and the bridge (see fig. 3.1 for reference). However, when a string is plucked it can ring for up to a few seconds and during such a note the player could manipulate controls with the plucking hand. The player's feet are unoccupied and could be used for controls but the double bass is quite heavy (~10-13 kg. depending on size and build) and is quite stationary which limits the reach of the player. Another

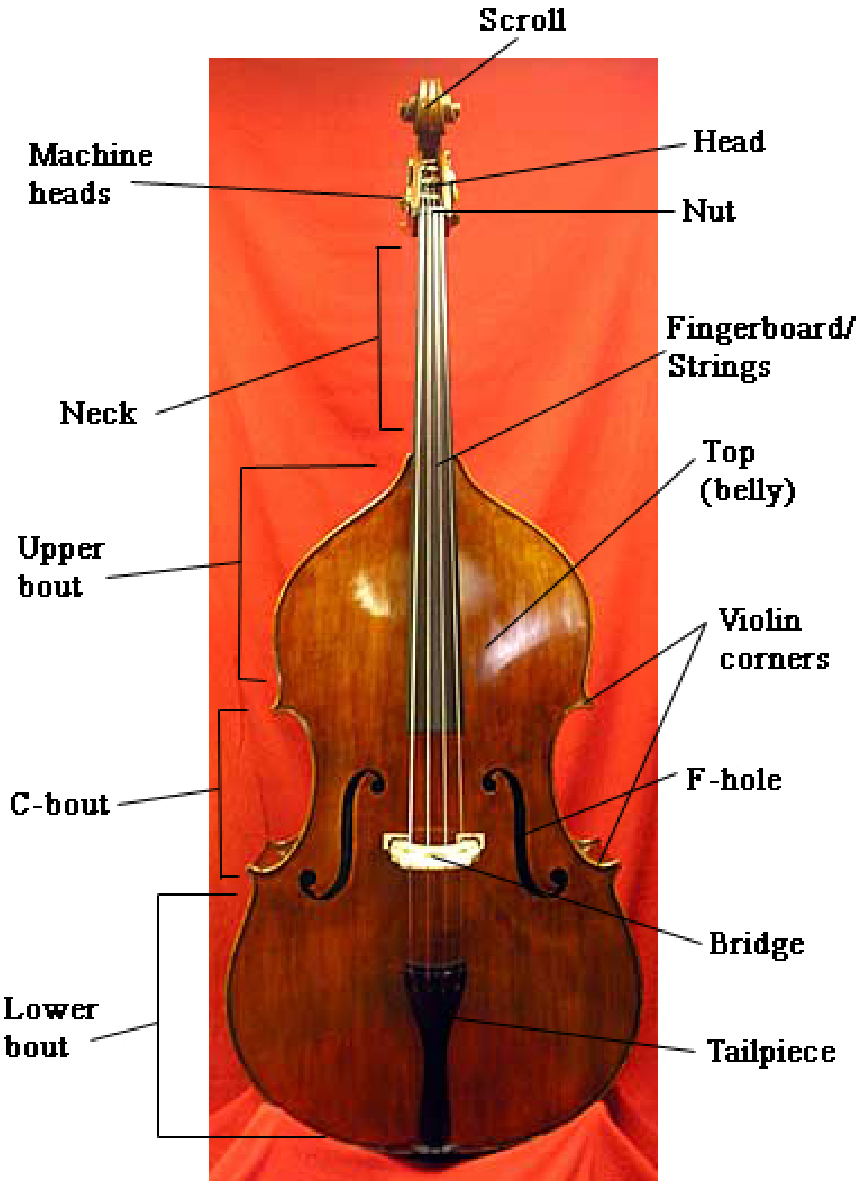

Figure 3.1: Principal parts of the double bass. [6]

option could be placing sensors on the bow e.g. buttons or IMU sensors. When gripping the bow it would be possible to control a few buttons. This idea was discarded, however, as it would require him to play using only the bow. Furthermore, an IMU sensor which could potentially disrupt his playing of the instrument. Considering the bandwidth we focused on the plucking hand and possible controls for it.

In order to generate ideas and explore possibilities, we made a Verplank [7] sketch (see fig. 3.2). The following is a quick summary of the Verplank sketch.

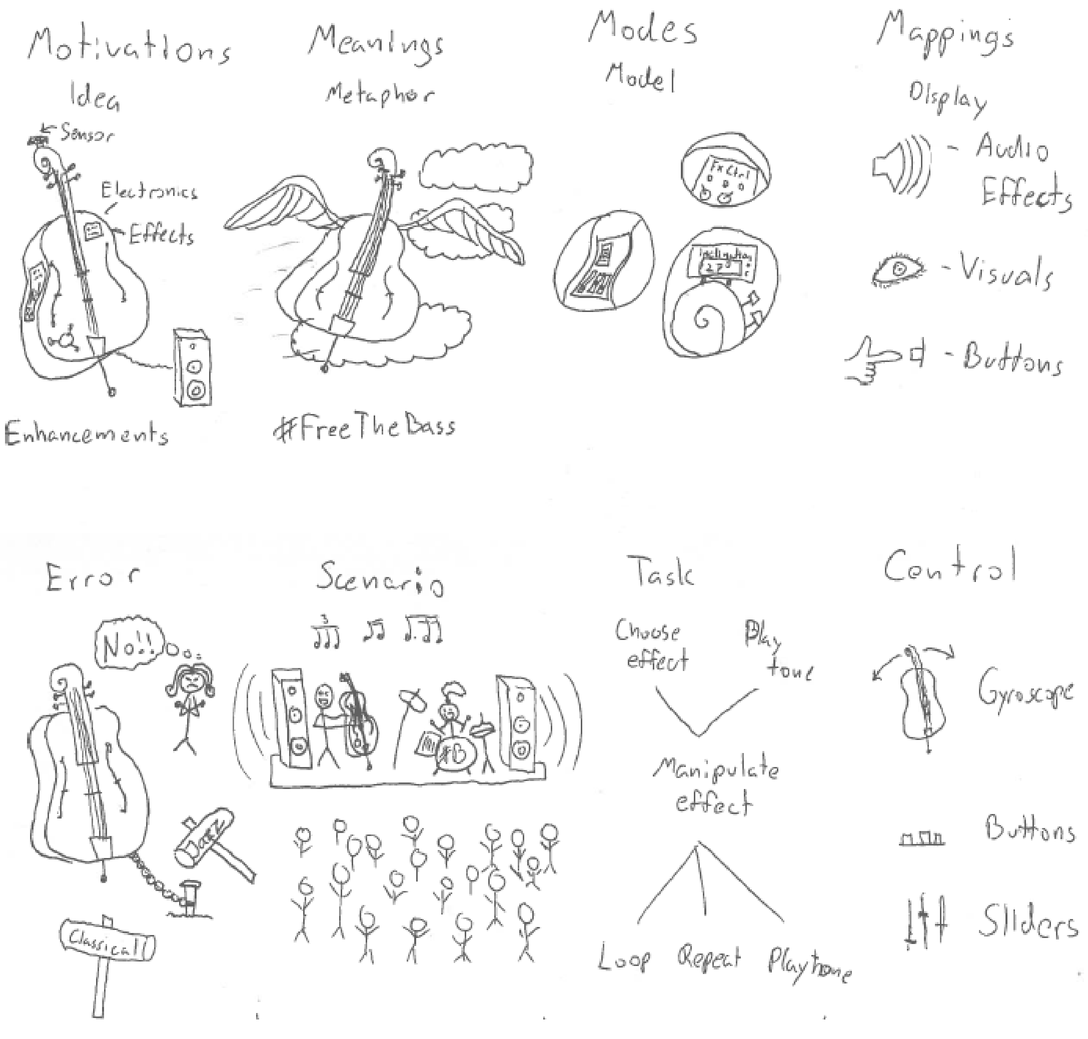

Figure 3.2: Verplank sketch showing the initial ideas for augmenting the double bass.

In "Motivations" an idea of extending the double bass and its inherent sonic features is depicted. This could be done through digital sound processing on a computer which is controlled by sensors. The motivations were sparked from the "Error", the idea that the double bass is, traditionally, only found in genres such as classical, jazz and folk music. Taking a look at "Meanings", the bass is depicted with wings, flying through the sky, meaning that we wanted to "free" the double bass and enable it to be a part of performances where you would not usually find it. That is, to enhance the role in untraditional genres or enable musicians to create entirely new genres of music. An example of the latter is depicted in "Scenario" showing a concert with only a drummer and a double bass; typically, the (double) bass has a supporting role in music, playing the fundamental notes in each chord - through our project the double bass could potentially be moved to the front of the scene, perhaps playing a soloistic role. The "Task" is to give the player easy access to and the ability to manipulate audio effects. This could also include the possibility of recording and playing back loops. In "Mappings" some of the various feedback possibilities through audio, visual and tactile are sketched. "Controls" of effects could be made available through sensors such as faders/sliders and buttons which are traditionally seen in sound and effect processors but possibly also a gyroscope for alternative controls.

3.2.1 Getting started

The desired sound effects were discussed with the collaborator using a mind map (see Fig. 3.3). He wanted a few common effects meaning reverb and delay. Other ideas

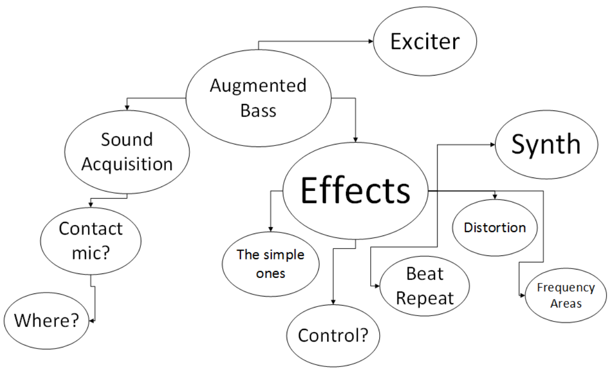

Figure 3.3: Mind map of various aspects to explore.

were explored such as having an effect that only affected a range of frequencies instead of the entire spectrum. This could for example be in conjunction with a distortion effect where only the higher frequencies are distorted. Since the double bass is naturally a very low frequency instrument, this would alleviate some of the problems with boomy sounds that entail if using distortion on sounds with heavy low frequency content. Being able to control a synthesizer alongside was briefly discussed but was quickly discarded as being beyond the scope of the project as it would shift the focus to building a synthesizer. Since the project was inspired by a previous project where contact microphones were used heavily, using them again was discussed. For the final choice of effects see section 4.1.

Before settling on the first iteration (see section 3.3) other ideas were considered. The first idea was to simply use a tablet, which would make it easy to design a user interface with the desired controls. However, the idea was discarded because the control unit would be placed on the front surface of the double bass. A touchscreen is hard to control when it is not visible as it has no tactile feedback. Another idea that was considered was the use of contact microphones to enhance the percussive elements that already exist in the double bass. One could, for example, drum a beat on the double bass and loop it. With signal processing, it would be possible to use the signals from several contact microphones to detect where the bass had been struck which in turn could be used as control signals. But this was also discarded as our collaborator wanted precise control of the effects.

3.3 Design Choices

The initial idea is sketched in fig. 3.4 where we had one row of buttons to activate and deactivate each individual effect. The other row of buttons would then select which effect to manipulate with the faders. This solution would allow the user to change parameters on any effect without having to turn it on or off. The sound is captured using an existing pickup on the bass, while the sensor data is sent to a computer for controlling the audio processing. An Arduino would be an ideal choice because it is easy to program making it ideal for prototyping.

Double bass players are limited in terms of time to use their hands for anything but playing the double bass. While they can turn on and off effects and change parameters between movements in music or really slow passages, if control of effects during constant plucking is necessary an alternative way of controlling effects is required. To this effect, an accelerometer was considered which could enable control of effects by leaning the instrument in one direction or the other. Notable challenges included the size of the instrument and the sheer weight of the double bass (~10 kg.). One is only able to lean the double bass forward so much before the angle impacts playability or before the weight of the double bass is too much to endure. A consequence of this would also be a lower resolution of the effect's min/max values. Another consideration was that the augmentation should impose as little as possible on the player's skill and play style. As Nicolls [8] notes in her study, subconscious movements naturally occurring in a performance could, if used to generate data for audio feedback, disrupt the artistic performance and change focus to controlling the sensors. Therefore, either a dead-zone in which the gyroscope sensor data would not be used or a switch to turn on IMU sensors would be necessary.

The placement of a control unit is very important considering the aforementioned

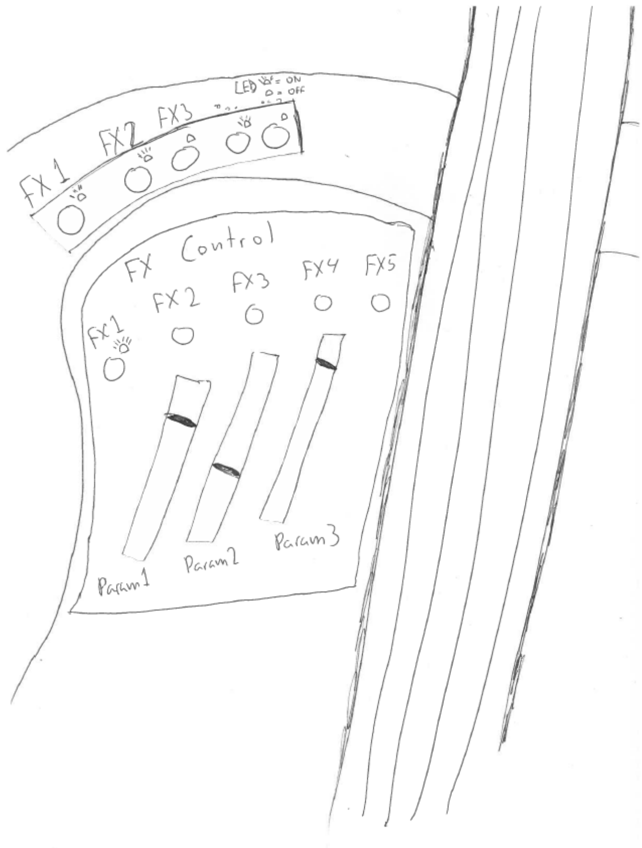

Figure 3.4: Sketch of the initial idea.

bandwidth and control space of a double bass player. The space immediately next to the plucking hand was considered. Through paper prototyping (see subsection 3.4.3), we established that this space (fig. 3.5) was the most easily reachable place on the double bass when not plucking or bowing because of the proximity to the strings. The interface itself should be easily used with just tactile feedback or at a glance down. Hence, the initial design features two rows of buttons and three faders. Each row should be on a different side of the bass and each button should have an LED lighting up when active. Three faders control the parameters of the most recently pressed effect. Precise and haptic feedback was necessary for the user to configure parameters without looking. With this in mind, motorised faders were used. Motorised faders were especially useful for a few different reasons. For one, as mentioned before, physical faders gave the user both haptic and visual feedback. Secondly, it meant that all parameters for a given effect could be saved and restored when switching between effects. This meant that the number of effects you could use with the box was limited only to the box size. It also meant we could fit a lot of effects into a single box. A feature that is an advantage when dealing with a double bass (see subsection 3.3.1).

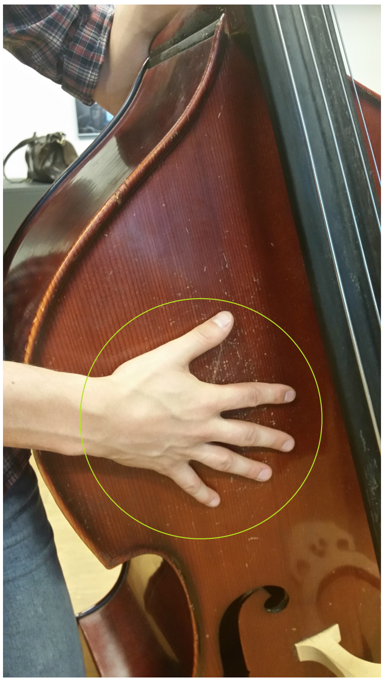

Figure 3.5: Control space for the UI.

To summarise, we needed to pick sensors that would enable the user to manipulate a selection of effects and the effect parameters. The controls should be placed in a specific place and should not require visual feedback to operate.

3.3.1 Foot Pedals versus Hand Controls

As mentioned in the introduction, it was a major change to move effect control from the established foot pedal to a hand controlled effects box. The obvious argument against hand controls was that the player needed free hands to play their instrument. However, if control over a high number of effects was wanted, having three pedals for every effect (assuming three parameters per effect) would take up a lot of space. And because of the double bass, any movement would be heavily impeded, especially, when actively playing. This means that, unlike guitar players who can move around on stage, double bass players cannot as easily operate a large array of effect pedals. Hand controls would enable the user to to finely control three parameters for any number of effects on a relatively small controller.

3.4 Testing

The ever-present dilemma in evaluating NIMEs and projects like ours is sufficient quantitative data versus appropriate time given to the performer to learn the instrument. As Dahlstedt [4] mentions; a musician accumulates experience and skill over many years of playing an instrument. This kind of evaluation is important as "quantitative lab tests can hardly tell us if an instrument works out there, with expert co-players, in front of an audience".

3.4.1 Qualitative Test

Since this project is done with a collaborator in the form of a double bass player we can account for this. This is done through continuous dialogue throughout the entire project, and through longer evaluation periods using the "long game" [9] method which resembles the time it takes to learn a new piece of music, with a semi-structured interview at the end. The collaborator will also be able to use The Box over a longer period of time once bug fixing of the second iteration is done. This will ensure he can practise and is qualified to answer the final interview. The collaborator will also use the project for two different performances after which a final interview will take place.

3.4.2 Usability Test

A second experiment was done with the box itself running a double bass backing track. It uses context-based tasks as described by Modhrain [9]. This test was conducted in both iterations and was done purely to test the usability of the box and the effects. The test was also helpful to verify the functionality of the prototype to make sure that the expert test would not be a waste of time. The first usability test sought to test whether or not the interface was intuitive. Intuitiveness in the box is important as the player only has a certain amount of bandwidth to play the instrument and use the box at the same time. The more intuitive the box is the more bandwidth the player has to employ it. In the second iteration, the usability test was more focused and made use of the System Usability Scale [10] to test usability.

3.4.3 Paper Prototype

A short paper prototype test was conducted before the first iteration was commenced. The purpose of this test was to evaluate button placement and placement of the box itself on the double bass and to test the idea of using buttons and faders to control effects. The paper prototype was comprised of three parts: activation buttons, select buttons and sliders.

3.4.3.1 Method

The test participant was presented with the paper prototype and received instructions on how to operate it. The user goals of the test were to have the user control all effects, change parameters and turn effects on and off. The user was then given a scenario in which he has to control the effects.

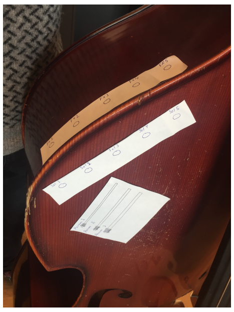

Figure 3.6: First UI placement configuration

"You're doing a performance and you need FX1 with parameter 1 and 2 at the maximum value and parameter 3 at the lowest." and then:
"FX4 needs maximum parameter 3 but parameter 1 and 2 set to their minimum values." and then:
"But now it's too much with FX1 so turn it off again." The test was tested on a fellow Sound and Music Computing student before bringing it to our collaborator.

3.4.3.2 Placement of the box

In order to find the ideal placement for the box, adhesive putty was applied to the paper prototype. Controls were placed on the upper bout of the double bass facing directly upwards (see Fig. 3.6). This configuration was less than optimal as it took too much time to reach up and use the buttons. The ridge buttons were moved down on the top (belly) of the bass along with the other buttons which was more appropriate. This led to the first iteration which can be seen in chapter 6.

Chapter 4

Choice and Design of Audio Effects

This chapter concerns everything related to the audio effects used in the project. The why of effect selection, the theory behind them and our implementation. It will also touch on various audio effect design choices made throughout the project. Due to the number of motorised faders, three parameters for every effect was the maximum where one fader was always used to control the dry/wet mix. Therefore, only two parameters will be discussed in the various audio effect sections.

4.1 Audio Effect Selection

Five different effects were chosen in collaboration with the bass player. The following chapter describes the choices behind the effects as well as the theory and implementation.

The first effect is the delay: beyond being able to use delay to echo notes and small phrases, delay can also be used for looping if the feedback is set at a high enough level (feedback=~1).

The reverb was chosen as the second effect to increase the vastness of the double bass. After a string on the bass is plucked, the energy rapidly diminishes and with it the note that was played. Reverb effectively increases the time the note can ring. The reverb also enhanced the effect of any other effect applied.

The third effect was a convolution engine. Convolution engines are commonly used to simulate the reverberations by mixing a clean signal with the impulse response of a specific room or place. While this project does use room IRs it also uses small recordings of running water, chainsaws etc. which gave very interesting sonic results.

The fourth effect was the beat repeat. The beat repeat effect essentially changes the way the double bass plays by adding rhythmical patterns to the produced sound. The beat repeat makes instant and periodic changes to the output level making the bass sound more like a synthesiser.

The fifth and final effect was the pitch shift. The double bass is a very low frequency focused instrument. The player is able to produce high notes but it requires a very precise technique and high skill level. Adding a pitch shifting an octave or two up enables the player to achieve high notes but by still playing them in lower octaves. The pitch shift effect can also could a note shifted a fifth up resulting in an open chord. The pitch shift effect was also a major tool to use for bringing the double bass into the foreground and out of the rhythm section.

Our collaborator wished to keep the selection small to keep things simple based on the idea that limitation sparks creation. Instead of implementing a large number of effects, only five were chosen. However, even by combining just the five effects, remarkable resulting sounds were achieved.

4.2 Pitch Shifting

In this project, pitch shifting plays an important role. It allows the performer to expand from the bass role and create entirely new compositions. Aside from the effect alone, pitch shifting is also necessary for the rhythmic processing effect (see section 4.7).

When designing a pitch shift effect it is important to consider if preservation of timbre is important? When a musician transposes their instrument (or to an even higher degree, voice) a different register is used and therefore also different timbrel features. If one looks at the singing voice, the timbre is largely determined by vocal cavities. These cavities emphasize parts of the spectrum which are called formants [11]. A singer has the possibility to control both the vocal chords (i.e. the frequency) and the formants (the timbre). As such, the design of a pitch shifting effect should consider both aspects. If only the pitch is changed but formants are constant you achieve pitch shifting with constant timbre. If the formants are changed but pitch remains constant you achieve something alike harmonic singing. Alter both independently and it starts to sound unnatural [11]. There are multiple methods of pitch shifting. The following subsections describe a few of them.

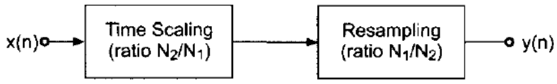

Figure 4.1: Pitch shifting by resampling and time scaling

4.2.1 Pitch Shifting by Time Stretching and Resampling

One method of pitch shifting is by resampling in the time domain. In the frequency domain, this is the same as compressing or expanding the sound over the frequency axis. The harmonic relations remain the same

$$f_i = i \cdot f_0 \tag{4.1}$$

but are instead scaled according to:

$$f_i^{\text{new}} = \alpha \cdot f_i^{\text{old}} \tag{4.2}$$

Resampling the sound also means altering the duration of the sound, therefore, the length must be rescaled as well as can be seen in figure 4.1 . The input signal goes from length N_1 to length N_2 followed by the resampling operation with inverse ratio N_1/N_2 followed by a reduction of length N_2 back to length N_1 [11].

Simply resampling a signal usually also introduces artefacts to the output sound such as clicks and pops. It also changes the formants and it usually renders the sound unrecognisable after shifting more than an octave [12].

4.2.2 Our Implementation of a Pitchshifter

Our implementation is based on delay-line modulation. A phaser controls a delay-line compressing or expanding the signal. This in itself would give pitch shifting but it also includes clicking artefacts. The artefacts occur when the phaser reaches zero. Therefore, the signal is windowed with a cosine function to smooth the signal when the phaser is at zero. This introduces a very perceptible phasing effect. To counter this, another delay-line run in parallel controlled by the same phaser is introduced. This delay-line is phase shifted 180 degrees and is cross-faded with the other delay-line to reduce the phasing effect. This is a relatively naive method as it takes no regard to formants and amounts of pitch, which makes the result of shifting sound artificial. See Fig. 4.2 for a block diagram describing the effect.

4.2.3 Enhanced Delay Line Modulation

Another method that is perhaps more suited for real-time processing is a block-based method. It uses an overlap-add scheme with three parallel time-varying delay lines. The signal is then cross-faded. Each block is read faster or slower according to which

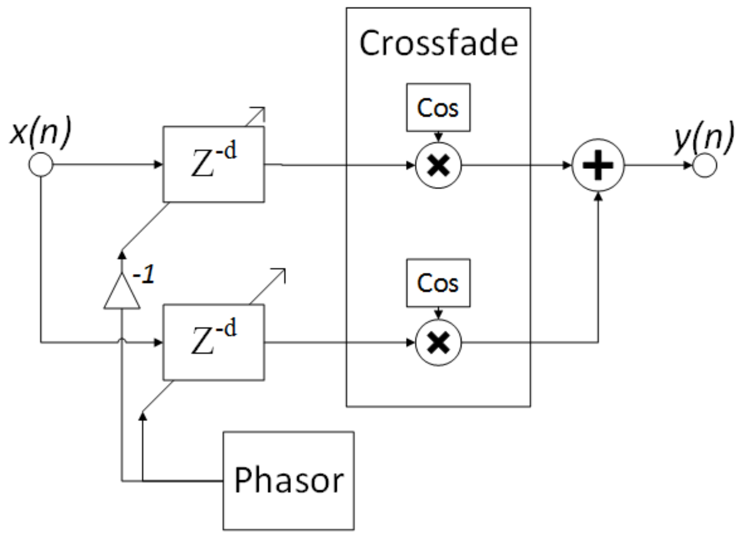

Figure 4.2: Block diagram of our pitch shift implementation

pitch is desired. The signal goes through delay-lines that are modulated by a sawtooth function followed by amplitude modulation and then summed [11]. Three cosine functions phase shifted 120°are used for modulation signals (see Fig. 4.3.) Artefacts that may occur are phasing-like effects which are an improvement from clicking and other local discontinuities of other methods [11].

4.2.4 Formant Preservation

Resampling and compressing/expanding an audio signal will change the pitch, but it will also change the formants. A technique to avoid this is the pitch-synchronous overlap-add method (PSOLA). Using this technique the short-time spectral envelope is resampled. The short-time spectral envelope is a frequency curve following the harmonic amplitudes (see Fig(4.4).

The harmonics are scaled according to

$$f_i^{\text{new}} = \alpha \cdot f_i^{\text{old}} \tag{4.3}$$

However, the harmonic amplitudes $a_i^{\text{new}} = env(f_i^{\text{new}}) \neq a_i^{\text{old}}$ are now decided by sampling the spectral envelope [11]. PSOLA is especially suited for pitch-shifting vocals, as it preserves formants. Pitch marks are found at the peaks of every period. These input segments hold the formant information and should not be changed. The distance between pitch marks determines speech period and this is what we want to change to shift pitch. A Hanning window is used to extract segments with a length of two periods. Not manipulating these segments means maintaining formant position. Overlap-add is then used to synthesise the pitch shifted signal by a factor of β. β is defined as the ratio of the local synthesis pitch frequency to the original one $\beta =$

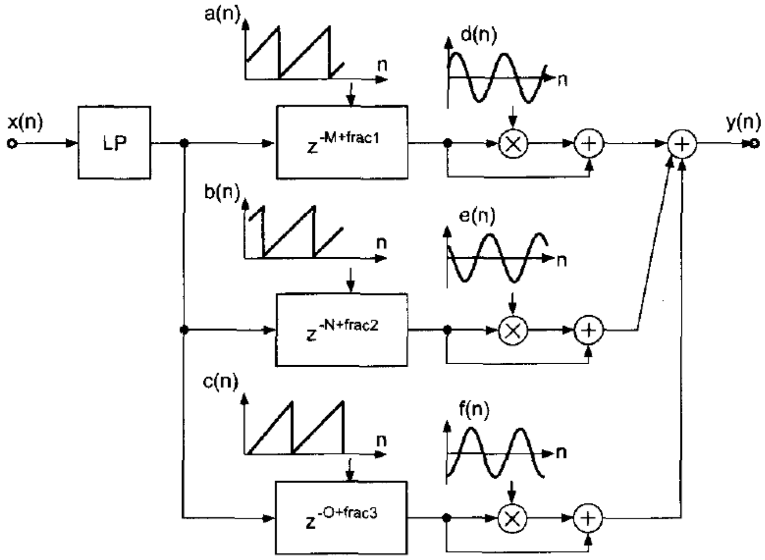

Figure 4.3: Block diagram of the enhanced delay-line modulation [11]

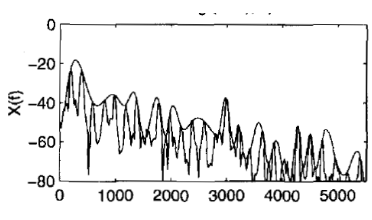

Figure 4.4: The short-term spectral envelope [11]

$\widetilde{f}_0(\widetilde{t})/f_0(t)$ the new pitch period is then $\widetilde{P}(\widetilde{t}) = P(T)/\beta$ [11]. When doing the overlap-add function some input segments are repeated when $\beta > 1$ for a higher pitch or discarded when $\beta < 1$ for lower pitch.

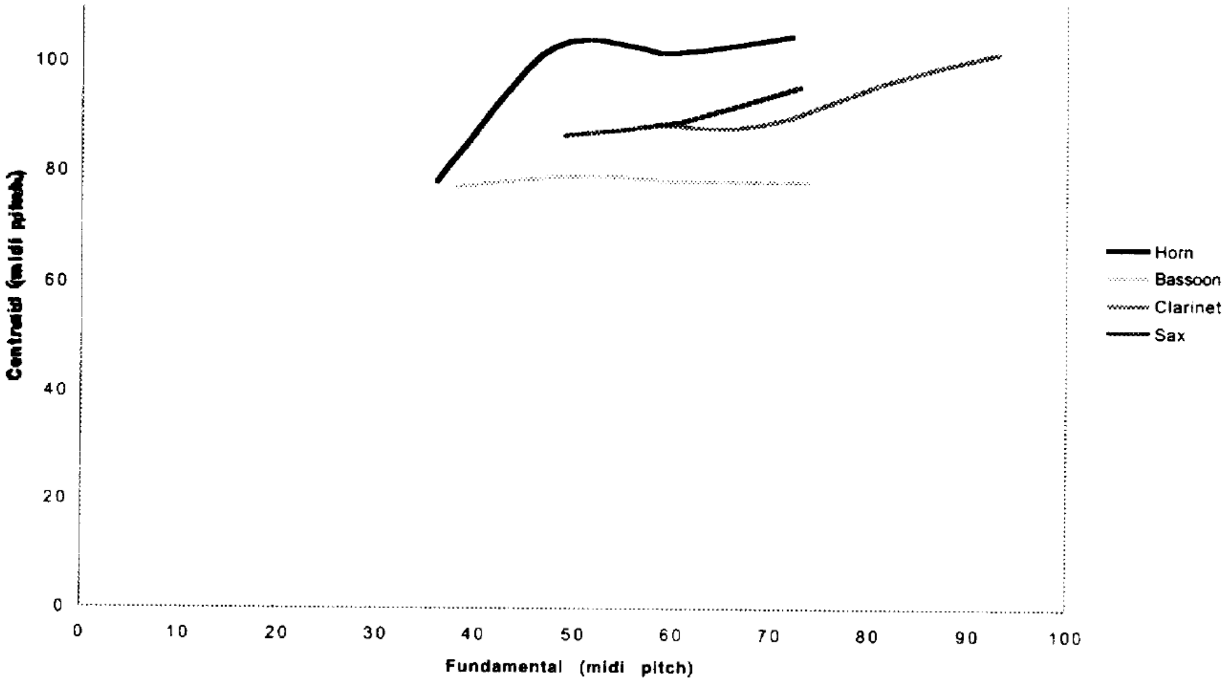

Figure 4.5: spectral centroid for various orchestral instruments [13]

4.2.5 Floating Formants

Using a formant preserving method to pitch shift is especially useful for vocal sounds as the vocal tract largely produces fixed formants for specific vowel sounds [13]. However, very few musical instruments have a fixed formant structure. As such, methods using formant preservation such as PSOLA (see 4.2.4) are not very suitable. On the other end, pitch shifting without regard to formants on an instrument is also far removed from the actual timbre of the instrument [13]. Plotting the spectral centroid as a function of fundamental frequency for orchestral instruments and the spectral centroid as a function of fundamental frequency for data transposed with no regard to formants, we see evidence for this.

The plots in fig. 4.5) and 4.6 show that the spectral centroid curve does not scale linearly. If we scale them non-linearly we would have transposed sounds with spectra that approximate the original sound. But this is more akin to frequency shifting and would yield inharmonic distortions. By using a hybrid method that uses the fixed formant transposition method and a non-linearly frequency scaling method we arrive at floating formant transposition. This method uses linear or polynomial curve inter-

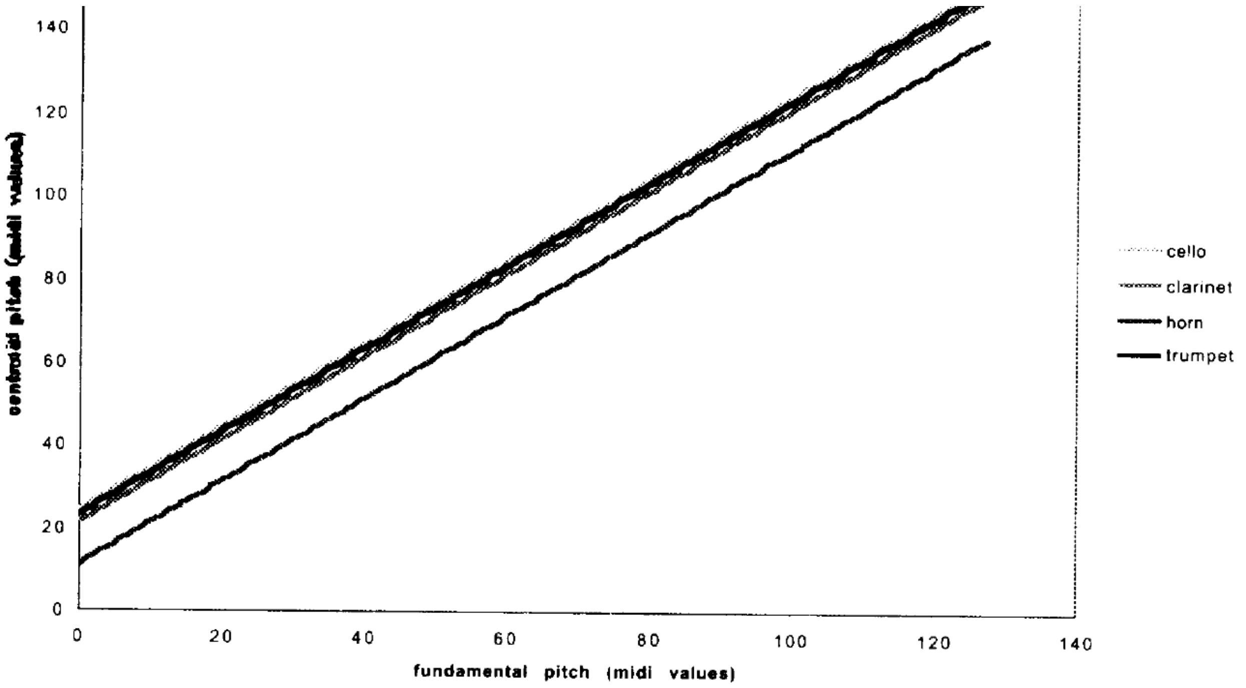

Figure 4.6: Spectral centroid for pitch shifted signal (no formant preservation)[13]

polation between amplitude peaks in the spectra to generate spectral envelopes [13]. We then fit the curve to a non-linearly frequency shifted spectrum but we only use it to derive a new spectral envelope. This can be applied to the transposed sound in the same way we saw a fixed formant spectral envelope used in PSOLA. Using these methods the object `gizmo` was made for Max/MSP.

4.2.6 Pitch-Shift Test

This assortment of pitch shifting methods leaves us with a choice. Which one is more suited for real-time processing and specifically for a double bass? A MUSHRA test [14] was used to investigate this question. The MUSHRA test was chosen because it is recognised as a very reliable method to measure audio quality through subjective listening [15]. A MIDI-file[16] was played using the MeatBass sample library for Plogue Sforzando [17]. Seven people were tested, aged between 20-30 years old and all students at Aalborg University. The test was conducted on a laptop running a `Matlab` script [14] with closed headphones. None of the participants was impaired on their hearing.

The MIDI-file was processed through the `gizmo` object (see subsection 4.2.5) and the delay-line modulation as implemented in subsection 4.2.2. Transposed versions of the MIDI-file were used as references for the pitch shifted versions. Three different MUSHRA tests were conducted. The original MIDI-file pitch shifted to an octave, a

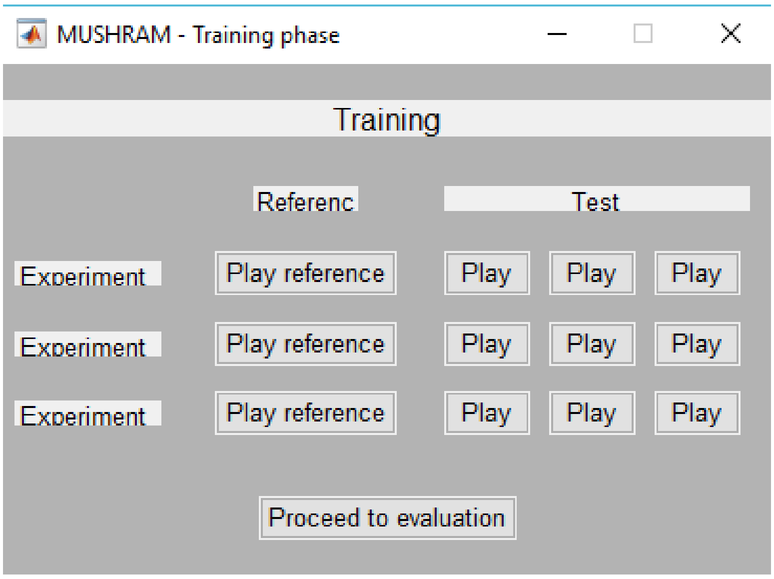

Figure 4.7: Training phase of the MUSHRA test [14]

fifth and two fifths. All tests had the reference sound and a reference sound lowpass filtered at 3.5kHz. The tests subjects were asked to go through a training period (see Fig.4.7) where they had access to all sounds to gain an understanding of a reference point to their ratings. After a break period corresponding to the training period, they were asked to rate the different examples according to how much the example approximated the reference sound (see Fig.4.8).

4.2.6.1 t-test

A t-test is used to analyse the data from the results. There are different types of t-tests. The most used ones are the one sample t-test, two sample t-test and the paired t-test. One sample t-tests were used to test if the mean of a single population is equal to a target value. This could, for example, be is the battery time on a given type of phone equal to 12 hours. The two sample t-test is used when you have two independent populations and want to test the mean against a target value. This could be is the battery time of phone type A greater than the battery time of phone type B. Finally, the paired t-test is used when you have a single population and wish to test the mean of the differences of dependent or paired observations against a target value. T-tests assume normality but results can still be valid if the data is not abnormally skewed [18].

For this specific test, we wanted to find out if the difference between the two means were significantly different.

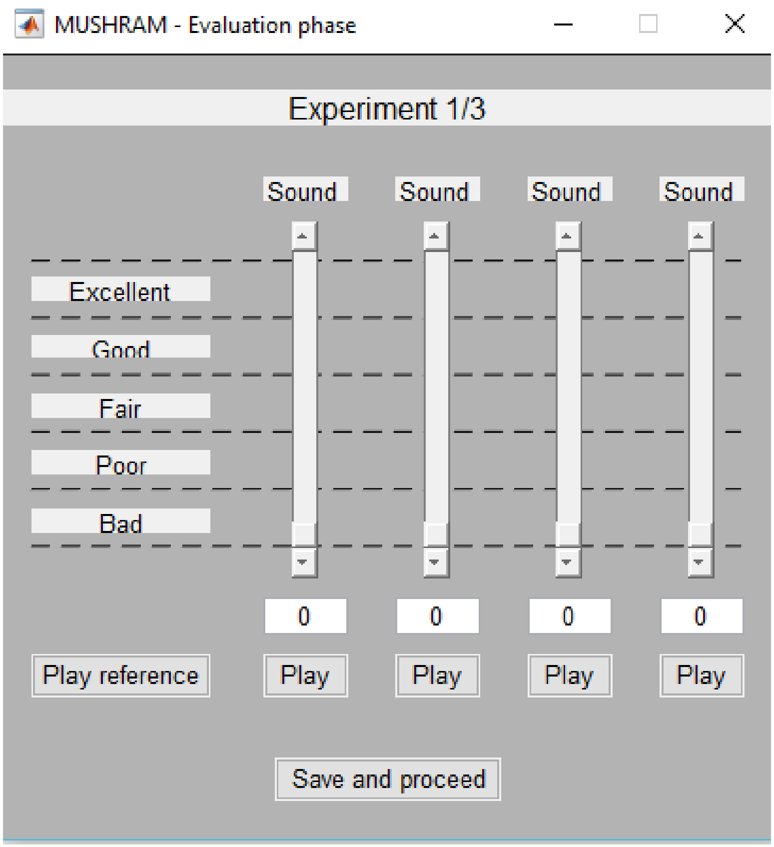

Figure 4.8: Testing phase of the MUSHRA test[14]

Table 4.1: T-test results

Pitch-shift	Octave	Fifth	Two fifths
t-test h	h = 1	h = 0	h = 1
t-test p-value	0.0019	0.5235	0.0094

Practically paired t-tests are done by using one sample t-tests on the difference in means. This can be done by using the statistic:

$$\frac{t = \bar{x} - \mu 0}{s/\sqrt{n}} \tag{4.4}$$

Where $\bar{x}$ is the sample mean, s is the sample standard deviation and n is the sample size.

The MUSHRA test is inherently designed as a within-subject experiment, meaning that the participants all get all the treatments. I.e. you need fewer participants to get a significant result as any outlying answers from a specific participant will answer with the same bias on all the treatments, this gives less error variance. The weakness is the carry-over effect. This could be a consequence of fatigue or training. Treatment one will be normal but the for the next treatments they will have trained and their answers may differ than if they had not. This is less of a problem for the MUSHRA test, as we assume test subjects are, if not expert listeners, then at least practised listeners. Fatigue might be an issue if the test is particularly long.

4.2.6.2 Pitch Shift Test Analysis and Result

A paired-sample t-test was used to analyse the test results (see figure 4.9), comparing the scores of `gizmo` and SMC (delay-line modulation). The null-hypothesis being the mean difference between the observations equal zero. As can be seen in table 4.1, there is a significant difference in both octave and two fifths with a 5% significance level. Pitch shifting only a fifth, however, produces results that cannot reject the null hypothesis.

One test participant chose the lowpassed reference over the reference in the TwoFifths test which means we could reasonably discard this person. But as we have few test participants we deemed this a reasonable test subject, still.
We can surmise the reason why there was no significant difference in the second test, Fifth. As can be seen in Fig. 4.6, the higher the pitch-shift factor is the more formants change when using a pitch shifting method without accounting for formants. This means we move further away from the reference sound which resulted in a low score in the test.

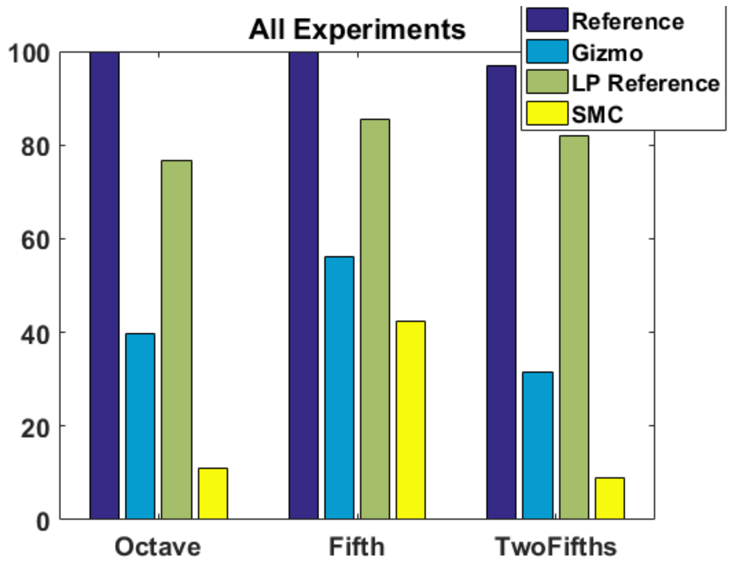

Figure 4.9: Pitch shifter test results

Based on these results, we used the `gizmo` object to pitch shift as it approximates an actual double bass at higher pitch shift factors.

4.2.7 Max Implementation

In the Max project, the two parameters are control of an octave and a fifth. Four different settings are available: -1 (1 octave/fifth down); 0 (no pitch-shift); 1 (1 octave/fifth up) and 2 (2 octaves/fifths up). In a previous iteration, the user-controlled the octave and fifth as well as the fourth by controlling the amount of each shift respectively using the three faders. See section 6.5 for more details on the change. In Max, depending on position, the fader sends a message to the `route` object that sends the correct MIDI-number to the `transratio` object that calculates the correct transposition ratio. For example, the MIDI-value 72 (12 semitones up from 60 where 60 corresponds to no pitch-shifting) is an octave up which would be the transposition ratio of 2. This is sent to the `pfft` object (the default object to process FFTs) with a window of 4096 and an overlap factor of 8. Inside the `pfft` object is the `gizmo` object (see section 4.2.5).

4.3 Delay

The delay effect is a simple but very powerful effect. In this project, the delay effect is an IIR feedback filter, essentially a comb filter. Using Eq. 4.5 the correct amount of delay is determined where τ is the delay in ms and *delay rate* is defined as the beat in a measure. I.e., a delay rate of 0.5 means the effect would repeat the signal in eighths

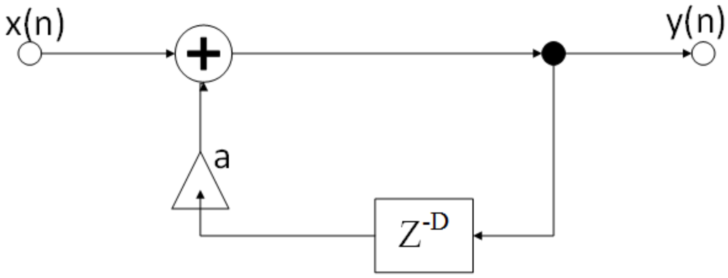

Figure 4.10: Diagram of the delay effect

and 0.25 would be sixteenths.

$$\tau = delayrate \cdot \frac{60000}{BPM} \tag{4.5}$$

The feedback factor (a in figure 4.10) is limited to $a \leq 1$ to ensure stability.

4.3.1 Max Implementation

The two parameters are the feedback factor and the delay rate. Feedback factor as mentioned above is limited to >=1 to ensure stability. It is, however, still possible to reach 1 in the cases where an infinite loop is desirable. This could be used to create a drone for the music piece. The decay rate parameter has 6 different settings. Half notes, fourths, dotted fourths, eights, dotted eights and sixteenths. The fader controlling feedback is simply scaled and controls the feedback factor object (a multiplication). The fader controlling delay rate is routed and sends a message depending on fader position, making use of Eq. 4.5.

4.4 Convolution

Traditionally, convolution is used to recreate the reverb of a specific location such as a church or a concert hall. This is done by taking the impulse response (IR) from the location and convoluting with an input signal. The IR could be gained by simply making a loud clap at the location and recording it. A more thorough way of obtaining it would be to use a sine sweep that goes through all frequencies and gives a complete IR. Another way of utilising convolution is through a rhythmically dynamic IRs. This could be the sound of pouring water into a glass, leaves rustling in the wind etc. Convolution can also be used to simply change the timbre of the sound. The sound of bees, while not especially interesting rhythmically, can still change the input signal significantly. The convolution effect is based on previous work by the authors. It makes use of the Max object `multiconvolve` from the *HISSTools IR Toolbox* [19]. They

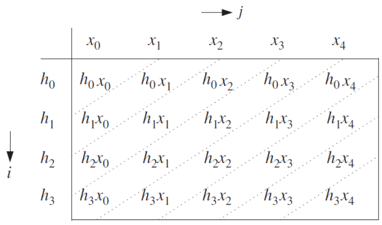

Figure 4.11: Convolution table [20]

use two different implementation techniques: time-domain based in the first half of the IR and an FFT-based partitioning scheme for the last half.

4.4.1 Time-Domain Convolution

The first technique, time-domain convolution is the most direct form and can be written:

$$y(n) = \sum_{\substack{i,j \\ i+j=n}} h(i)x(j) \tag{4.6}$$

, where h is an impulse response and x is the clean signal. Consider the example put forth by Orfanidis [20]; an order-3 filter and a length-5 input signal. The filter, input and output blocks would then be:

$$h = [h_0, h_1, h_2, h_3] \tag{4.7}$$

$$x = [x_0, x_1, x_2, x_3, x_4] \tag{4.8}$$

$$y = h * x = [y_0, y_1, y_3, y_4, y_5, y_6, y_7] \tag{4.9}$$

Where asterisk($*$) is defined as convolving. The example above uses a smaller sample. We can then use a convolution table to calculate output. We fill the table by multiplying the nth x element with the corresponding y element. We fold the table anti-diagonally and sum the strips to form the output. This is an FIR method of convolution using just a single block of samples

4.4.1.1 Overlap-Add Convolution in Time-Domain

When handling an infinite input, however, you do not have a single block of samples you can conveniently convolve. As such, we need another approach. This approach could be using an overlap-add block method. The input data is divided into smaller blocks which are then convoluted with the IR. These are then put together again (see Fig. 4.12. Each output block will be longer than the input block and the overlapped portions are added together.

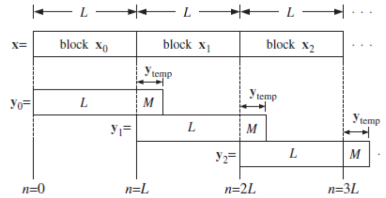

Figure 4.12: Overlap-add table [20]

4.4.2 Frequency Domain Convolution

While it is possible to do convolution in the time domain, it is much more efficient in the frequency domain. One method of doing convolution in the frequency domain is the circular convolution.

4.4.2.1 Circular convolution

Circular convolution is simply where the DTFTs of IR h and input signal x are multiplied[20].

$$y = h * x \leftrightarrow Y(\omega) = H(\omega)X(\omega) \tag{4.10}$$

We can then regain $y(n)$ by taking the inverse DTFT of the product of the two DTFTs [20].

$$y(n) = \int_{\pi}^{\pi} Y(\omega)e^{j\omega n}\frac{d\omega}{2\pi} = \int_{\pi}^{\pi} e^{j\omega n}\frac{d\omega}{2\pi}H(\omega)X(\omega)\frac{d\omega}{2\pi} \tag{4.11}$$

However, simply taking $y = IDTFT(DTFT(h) \cdot DTFT(x))$ is not practical, as ω integration requires of knowledge of $Y(\omega)$ at a continuous range of ω's [20]. Instead we replace all the DTFTs by N-point DFTs, or a fast version with FFTs.

$$\tilde{y} = \widetilde{h * x} = \text{IFFT}(\text{FFT})(h) \cdot \text{FFT}(x)) \tag{4.12}$$

For the circular convolution $\tilde{y}$ to assent with convolution y, DFT length must be at least the length L_y of the sequence y [20] where $L_y = L + M$ for a signal x of length L convolved with an order-M filter h. thus

$$\tilde{y} = y \text{ only if } N \geq L_y = L + M \tag{4.13}$$

the vectors h and x have lengths less than N Eq. (4.13) and so we zero pad them at their ends before computing the FFTs. if $N < L_y$ part of the tail of y wraps around the beginning part of y [20].

4.4.2.2 Overlap-Add Convolution in Frequency-Domain

When dealing with infinite inputs, we run into the same problem as we did in section 4.4.1.1. We cannot reasonably process the entirety of an infinite signal. Once again we have more practical solutions such as overlap-add. The condition (4.13) can not be fulfilled either as the length $L \to \infty$. Instead the overlap-add method discussed in 4.4.1.1 can be used. Circular convolution is still used on the input blocks. The FFT length must still satisfy Eq. (4.13) which also means we can determine the length of the input segments.

$$L = N - M \tag{4.14}$$

The output blocks and then be computed by

$$y_0 = \tilde{y}_0 = \text{IFFT(FFT)}(h) \cdot \text{FFT}(x_0)) \tag{4.15}$$
$$y_1 = \tilde{y}_1 = \text{IFFT(FFT)}(h) \cdot \text{FFT}(x_1)) \tag{4.16}$$
$$y_2 = \tilde{y}_2 = \text{IFFT(FFT)}(h) \cdot \text{FFT}(x_2)) \tag{4.17}$$
$$\tag{4.18}$$

and so on.

4.4.3 Max Implementation

What is controlled in the convolution effect is the sound used to convolute the signal with. One fader chooses one out of three different sounds and the second fader chooses which bank (three new sounds, chosen with the first fader) is used. This enables the user to choose between 9 different sounds to convolute with. They are as follows: Water being poured into a glass, metal railing being struck, a bee swarm, water bubbles, a chainsaw, a spinning coin, random digital noises, a typewriter and lastly water trickling through a drainpipe. These were chosen through subjective listening by the authors trying out a multitude of sounds. The intention is to form a well-balanced sound bank that includes both more classic IR's such as the metal railing but also the more dynamically interesting ones such as water poured into a glass.

Using a fader is not the optimal solution in terms of precision when choosing a certain sound. Therefore, only three options per fader are used. This gives the user two fixed points they can always hit (both ends of the fader) and the midway point. Visual feedback in the form of an RGB LEDs changing colour also helps with this task. The faders are also used for all the other effects for which they are more intuitive. We sacrifice some precision for being able to reuse the faders for everything, saving a lot of space.

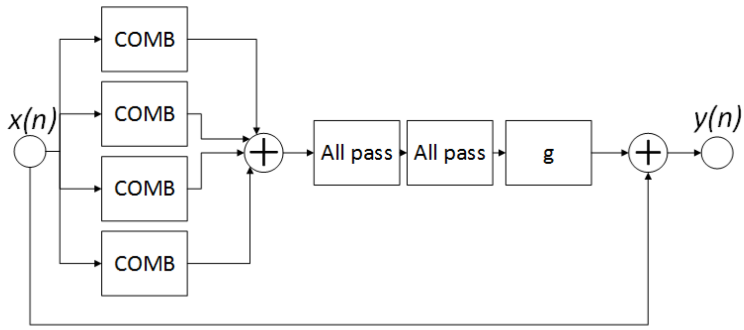

Figure 4.13: Schroeder's reverb

4.5 Reverb

A reverb effect seeks to simulate the reverberations that naturally occur in a room when sound is propagated inside [11]. This effect is a staple among musicians. There are multiple ways of realising this effect.

4.5.1 Schroeder's reverb

The typical model consists of comb filters in parallel followed by all-pass filters in series (see Fig.4.13). The comb filter (see Fig.4.10), so called because of its comb-like frequency response, serves as the initial echoes that are less dense while the all-pass filters in a series act as the later echoes that have high densities. The rule of thumb by Schroeder is, that a natural sounding reverb requires about 1000 echoes per second [21]. With a delay of 0.04 seconds, the comb filters each give 25 echoes. This equals 100 echoes for the comb filters if three all-pass filters (see Fig.4.14) are added in series they effectively multiply this number by 10, giving us 1000 echoes per second. The all-pass filters have a flat frequency response but if the delay time of these is more than 50ms (the integration time of the ear [11]) the time domain qualities of the filter become more apparent. This can result in a metallic timbre. In Moorer's reverb, he suggests setting the all-pass delay length to 6ms and the coefficient to 0.7 to combat this. To reduce superimposition of echoes delay lengths should be mutually coprime.

4.5.2 Rev3 - an External Reverb

This project made use of the `rev3` object by Mitch Turner [22]. We chose to use an external reverb instead of developing our own. Designing a realistically sounding reverb could easily be an entire project in itself which was not within the scope of this project. This particular reverb was frequently mentioned in Cycling'72's user-forums and chosen based on subjective listening by the authors. It was a fairly hard task to reverse-engineer the `Rev3` object. It has 16 delay lines with varying degrees of delay ranging from 10 to 130.50199 reducing superimposition of echoes. Four low-

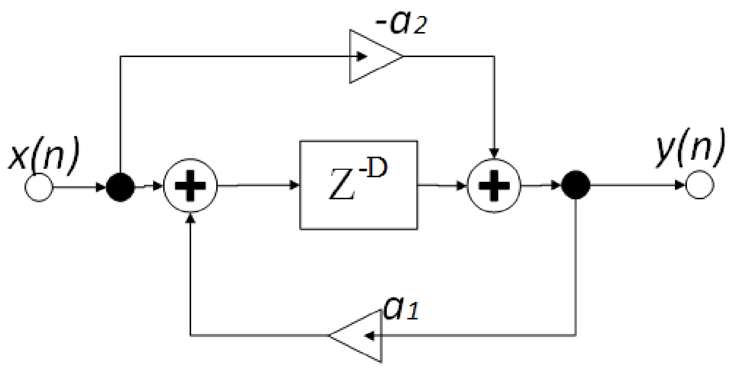

Figure 4.14: All-pass filter diagram

pass filters are used to ensure faster decay time in the higher frequencies. All delay lines continuously mix with the other delay lines either by adding or subtracting two signals (see appendix F).

4.5.3 Max Implementation

The two parameters are the feedback factor and dampening. Both are continuous values going from 0 to 1. This is overall easier to control than discrete values, such as the parameters of the convolution effect. The first fader controls feedback and the second fader high-frequency dampening. There was also the choice of giving control of the cross-over frequency

4.6 Beat Repeat

The beat repeat effect is used to add specific rhythmic qualities to the input signal based on the pattern set up by the user. The effect might be wrongfully named but it was the term our collaborator continuously used. Our beat repeat effect is basically a pattern-controlled volume gate. The pattern is input using the `multislider` Max object which a list of values; in our case float values from 0-1. There are 32 values corresponding to 4 bars of 16th notes. The list is traversed in tempo corresponding to the overall BPM and the values are used to control a gain slider(see fig. 4.15). An example of pseudocode to traverse an array consisting of the beat repeat values input by the user:

```
for (i = 1; i <= 32;) {
   value of position i in array = value of volume gain
    i = i+1
}
```

Where the speed at which `i` increases by 1 is equal to the BPM set by the user. The beat repeat effect's customisability allows for more rhythmically complex and interesting patterns than if it had been controlled by an LFO. The beat repeat effect is especially pronounced when using a bow on the double bass.

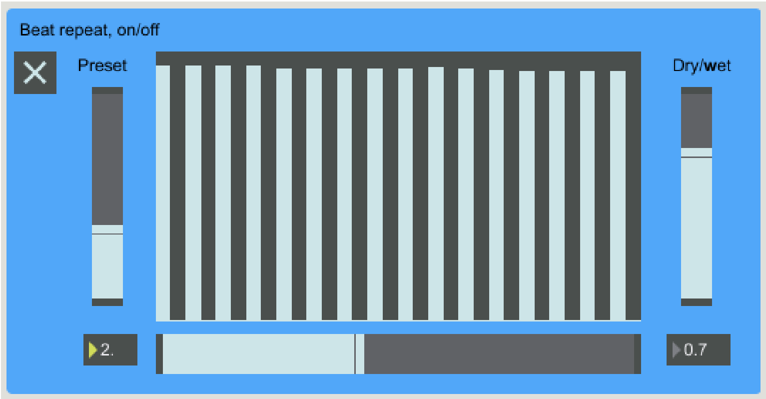

Figure 4.15: Beat Repeat effect in Max with an input pattern of alternating 1's and 0's in the `multislider` object

4.6.1 Max Implementation

Currently, only one slider is used to control parameters in the Beat Repeat effect. The first fader goes through different presets that either the user has set up on their own or use the default ones. The last preset takes input from TouchOSC [23] on a mobile phone. This enables the user to input their own pattern without having to go into the program itself on a computer (see Fig. 4.16). The second fader could be used to expand the effect to control different filters instead of amplitude only. Such filters could be bandpass to create a wah-wah effect or a highpass filter.

4.7 Rhythmic Processing

The Rhythmic processing effect is inspired by Diego Stocco's set-up in his Feed Forward Sounds series [24] in which the input sound (a guitar for example) is transformed into several tracks which include bass drum, snare drum, hi-hats and a bass track. This is done through filtering the various frequency bands needed for each track. Low-end frequencies for the bass drum and the high end for hi-hats and so on. Depending on the track, they also need effects such as erosion, saturation, compression to increase attack and enveloping to define proper beats in the rhythm. This also makes it heavily reliant on the performer keeping the exact tempo. Lag behind and you might miss the

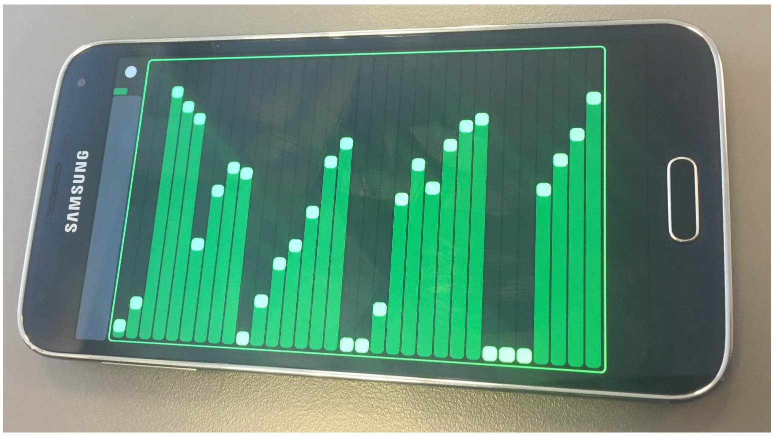

Figure 4.16: TouchOSC on a phone for a Beat Repeat pattern

bass drum timing.

The challenge in using rhythmic processing on a double bass is the input frequencies. Naturally, the double bass has more bass than treble and as such, any high-frequency content you can extract from it (if any) is ill-suited for music. The process also relies heavily on the high attack from the input sound. Striking the strings with a wooden rod or something similar is preferable to simply strumming with your fingers. A double bass, however, is diffuse in its sound. A notable exception is when you strike the body of the base to provide percussion.

The system must also have no lag. The biggest offender of this is the pitch shifting that is necessary to obtain hi-hats. As described above, it is lacking high-frequency content to give much sound for a hi-hat sound. This necessitates the use of a pitch shifter. The problem with pitch shifting is the delay inherent in the effect. The quality of the pitch shifting must come in second in order to ensure no noticeable delays.

A different approach is to use delay based rhythmic processing. Rather than enveloping the amplitude to create beats, you use a certain amount of samples of the input sound to loop. Different delay timings can create different kinds of beats. This also alleviates the dependency on keeping a perfect tempo as you essentially start the beat every time you strum the instrument.

At this time, unfortunately, the rhythmic processing system has been prototyped but not fully realised. The issue lies within the pitch shifter. At the moment we can not fully remove the delay which makes it a difficult process when relying so heavily on precise timings. A solution could be to measure the exact delay from the pitch shift and shift timings accordingly, but this has not been tried yet.

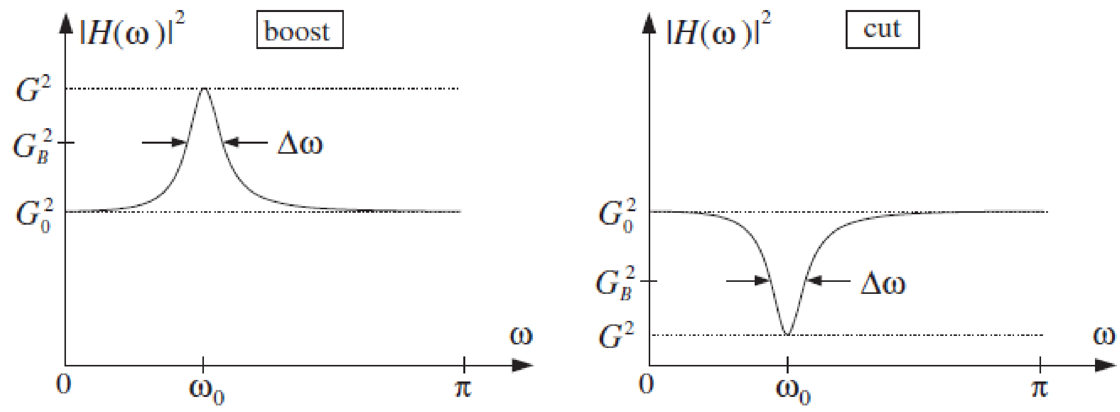

Figure 4.17: The difference between boosting and cutting [20]

4.8 Equalizer

Equalizing gives an important factor of control to the user. However, as the product is supposed to be used on stage, we reason that the sound technician on the venue will have control over any equalizing of the final sound. In the various effects, however, an equalizer is of much use. Therefore, what follows is an overview of equalization based on previous work by the authors. There are two main types of equalizers. Graphic equalizers where the frequency bands are divided into numerous parts with a fixed 3-dB bandwidth. Only the gain for the particular band can be changed. Parametric equalizers are more flexible in that you can change more parameters: Reference gain G_0, gain at center-frequency ω_0 and width $\Delta\omega$ at a fitting G_B that is between G_0 and G[20] as can be seen in figure 4.17. There are various ways of calculating a fitting G_B. Orfanidis [20] uses the example of putting $\Delta\omega$ equal to 3-dB width. This could mean 3-dB below the peak ($G_B^2 = G^2/2$) or 3-dB above the reference ($G_B^2 = 2G_0^2$). Both alternatives require that $G^2 > 2G_0^2$ which means boost gain has to be 3 dB higher than the reference. When $G_0^2 < G^2 < 2G_0^2$ it means any G_B^2 that lies in $G_0^2 < G_B^2 < G^2$ works.

Orfanidis [20] mentions another choice which is to take the arithmetic mean of the end values:

$$G_B^2 = \frac{G_0^2 + G^2}{2} \tag{4.19}$$

This Eq.(4.19), is the way we chose to implement it.
Orfanidis [20] defines the transfer function for a parametric equalizer as:

$$H(z) = \frac{\frac{(G_0+G\beta)}{(1+\beta)} - 2\frac{(G_0 cos\omega_0)}{(1+\beta)}z^{-1} + \frac{(G_0+G\beta)}{(1+\beta)}z^{-2}}{1 - 2\frac{(cos\omega_0)}{(1+\beta)}z^{-1} + \frac{(1-\beta)}{(1+\beta)}z^{-2}} \tag{4.20}$$

With the parameter β defined as:

$$\beta = \sqrt{\frac{G_B^2 - G_0^2}{G^2 - G_B^2}}\tan(\frac{\Delta\omega}{2}) \tag{4.21}$$

ω_0 as:

$$\frac{2\pi f_0}{f_s} \tag{4.22}$$

and $\Delta\omega$ as:

$$\frac{2\pi\Delta f}{f_s} \tag{4.23}$$

4.8.1 Max Implementation

For now, the user had access to any equalisers. However, if that were to be changed, the user could control the filter type on one fader, Q on a second and center-frequency gain on the third. Equalising without any visual reference would be a rather cumbersome task and would be best be done before a performance.

4.9 Sonic Design

When creating or augmenting an instrument it is also important to consider what the instrument is going to sound like and how this sound is produced. This section discusses choices made in the project of the following: The specifics of the audio effects and the sound production of the entire instrument.

4.9.1 Audio Capture

The double bass is, first and foremost, an acoustic instrument. The sound is produced when the strings are plucked or bowed which causes the strings to vibrate. The vibration is transferred via the bridge to the body which makes the entire body resonate with the frequency of the vibration. To add audio effects to the double bass, the sound needs to be captured which can be done in a few ways. Our collaborator's bass is fitted with a "Realist" pickup [25] placed between the bridge and the body. The pickup consists of a piezoelectric disk (i.e. contact microphone) which turns the vibrations into an electric signal which is output through a 1/4-inch jack (see fig. 4.18). An-

Figure 4.18: A pickup for a double bass.

other option would be to use a microphone (e.g. condenser or dynamic). Placement of the microphone in relation to the instrument basically comes down to the taste of the player and/or sound technician and how they want the sound to be like. General guidelines [26][27] suggest aiming a microphone at the bridge or f-holes at a distance of 15-45 cm.

The advantage of using the pickup is that only the sound from the bass is captured for processing resulting in a very clean signal as only the vibrations are captured. However, the signal is a relatively weak one and might need amplification.

A conventional microphone can more clearly capture the full signal of the resonating body and details such as bowing, plucking or even hitting/slapping the bass for percussive elements. And although it requires much less amplification than the pickup, there is a risk of other instruments being caught be the microphone.

Throughout this project the pickup was used for its clean signal properties and because it required no extra set-up, enabling fast testing for prototyping purposes.

4.9.2 Audio Output

There are multiple considerations regarding outputting the captured sound from a double bass. Using a pickup, one option is to plug the double bass directly (or via a pre-amplifier) into an amplifier specifically tailored to the double bass, similar to amplifiers for guitars or electric basses. Another option is to plug the bass directly into a PA-system, which sends the signal to loudspeakers. In this project, the captured signal is sent from the microphone to the audio interface which passes the signal on for processing in Max/MSP. When testing the processed signal was output via an audio interface and sent to a PA-system.

4.9.3 Effects Chain

How effects are chained have a significant impact on the resulting sound. For example, reverb is traditionally put at the end of the chain. As the reverb effect creates a diffuse sound, effects that rely on a more clean signal for input will falter or give too much. This is apparent in the convolution effect where a higher amount of energy in the low-frequency range will cause the effect to output a much higher amplitude than intended. Pitch shift is set at the start since everything is intended to be shifted. Delay and convolution could be switched around but it is important that they come before Beat Repeat. The Beat Repeat is supposed to be very pronounced and during the first usability test (see section 6.6) people expressed that it was less than satisfactory when chained before convolution.

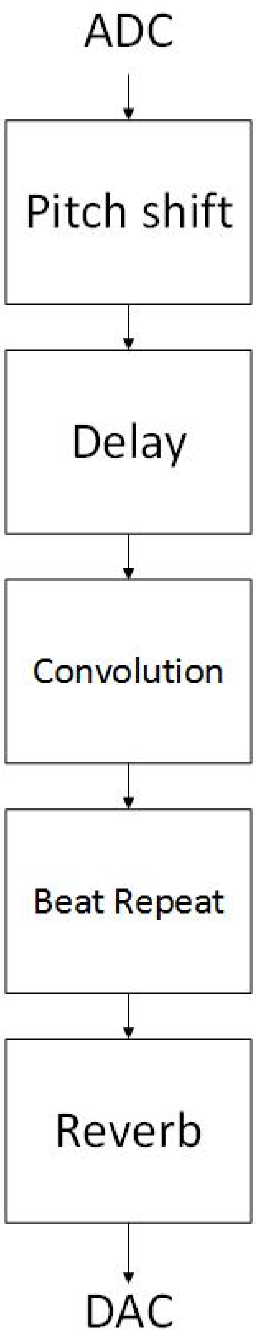

Figure 4.19: Diagram of the effects chain.

Chapter 5

Max - Software and code

This chapter concerns itself with the software set-up and code used to make the Max project. More specifically, it looks at different snippets of the code that were either difficult or play an intricate part of the software. Max/MSP is a visual programming environment in which all of the project's sound processing and sensor data mapping is handled. The project uses Maxuino [28] in combination with an Arduino to run all the processing via Max on a computer. This enables Max to send and receive messages from the Arduino.
See Fig.5.1 for a quick overview of the data flow in the program. The overview is not comprehensive but provides the most important blocks.

5.1 Set-up

In Fig. 5.2 the Arduino is automatically selected and the same `loadbang` is sent off to initialize all the pins as can be seen in Fig. 5.3. A `bang` is used to trigger events in Max/MSP and the `loadbang` sends a `bang` as soon as the project starts up.

In Fig.5.4 The data received from the Arduino is passed through a `route` object to filter analogue data from digital data and is, in the end, passed through outlets leading to both the main patch and the motor control patch.

5.2 Button Configuration

The data coming from the Arduino once a button is pressed is a stream of continuous 1's. We are only interested in the first bang from the button. Therefore, the other 1's are filtered out using the `onebang` object which only lets the first `bang` through until it has been reset by another `bang` (see Fig.5.5).

When selecting an effect (not to be confused with activating) to manipulate, only one should be active at a time. Fig.5.6 shows our way of implementing this system. A `toggle` sends out a 1 or 0, the `sel 1` object picks the 1 up and the `trigger 1 0` object

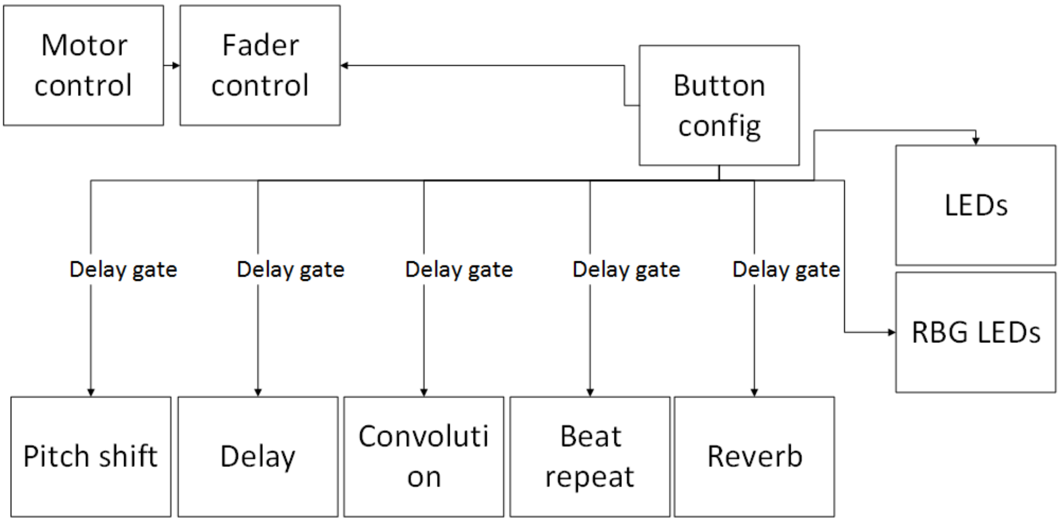

Figure 5.1: Diagram overview of data flow

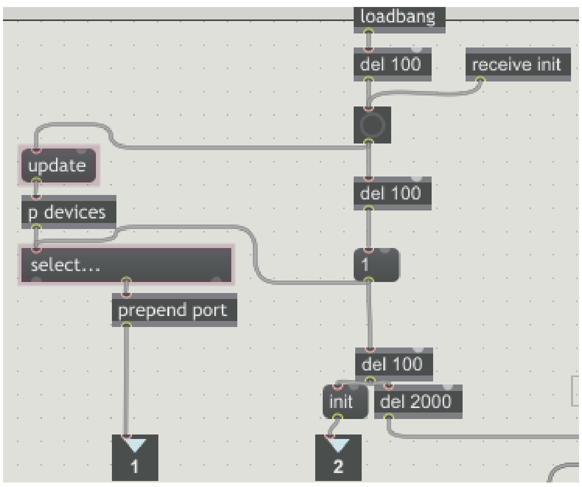

Figure 5.2: `loadbang` set-up

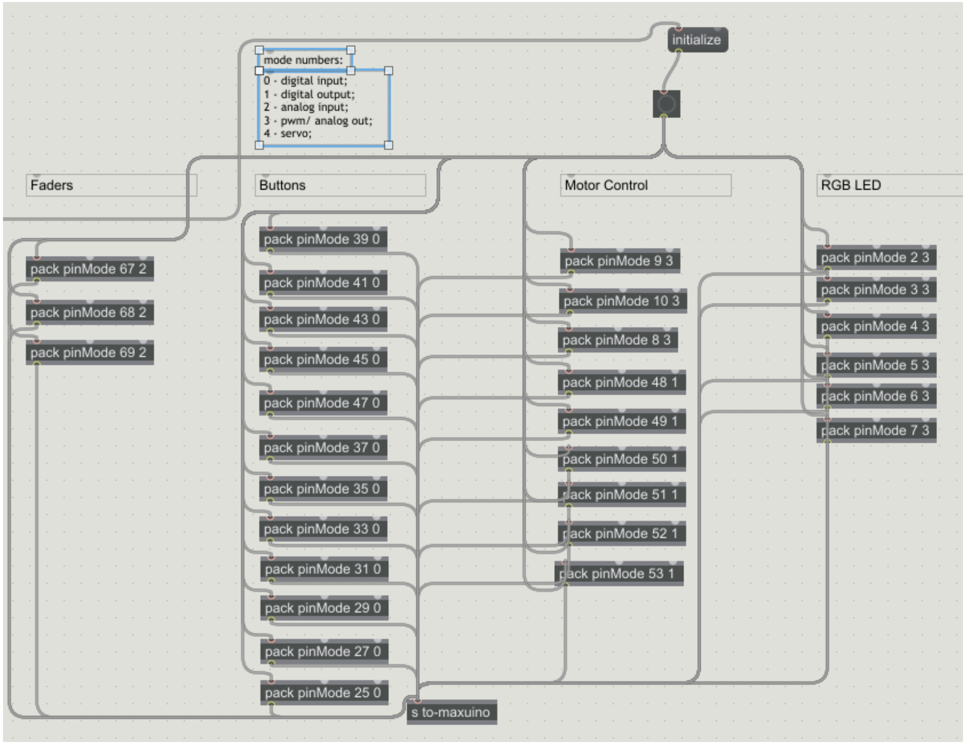

Figure 5.3: Initializing pins

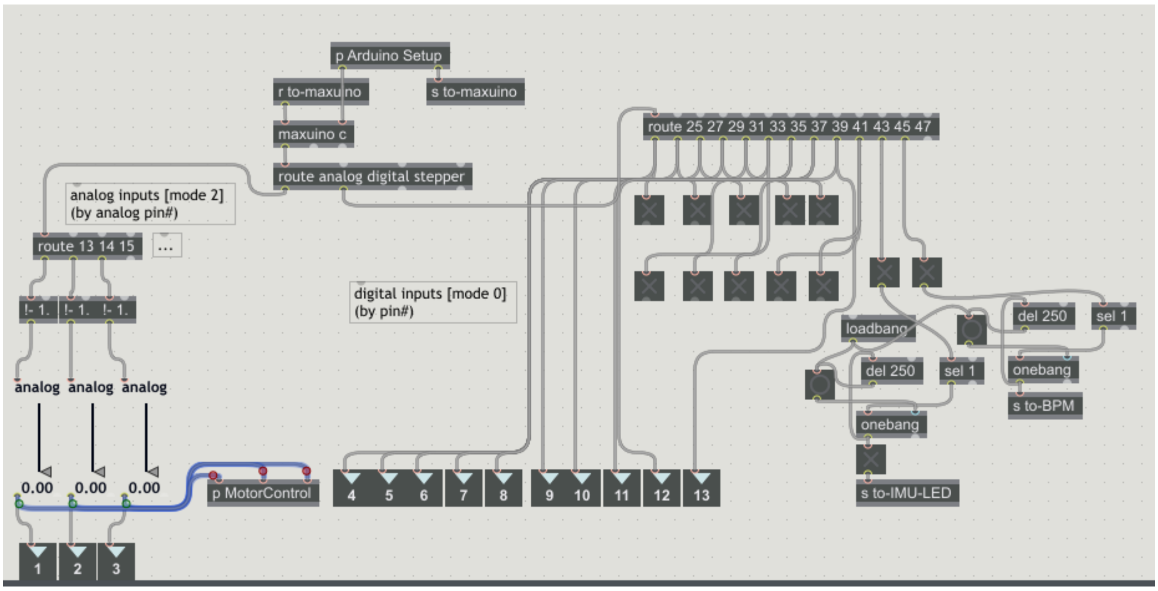

Figure 5.4: Passing data setup

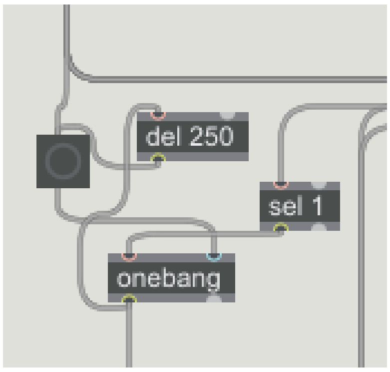

Figure 5.5: `onebang` to deal with continuous input

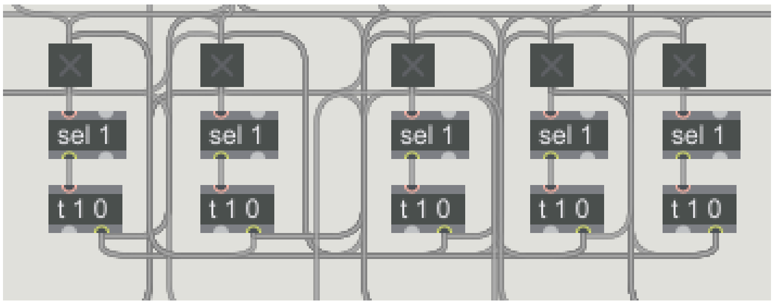

Figure 5.6: Effect activation logic

sends out a 0 to every other `toggle`.

5.3 Motor Control

Inside the motor control patch, three things happen: filtering the effect selection `bang` from the effect de-selections (see Fig.5.7); three motor control patches for each motorized fader (see Fig. 5.9) and, finally, a patch to save previous fader positions (see Fig.5.9).

In Fig. 5.7 effect selection toggle values are sent through to the motor patch. Here they are filtered. Selection is sent to the three individual fader patches that control motors, while de-selections are sent to the patch that saves previous fader locations.

In order to facilitate a fader that remembers its location for every effect, five of these patches (see Fig. 5.8) are in control of that. Every time an effect is de-selected a `bang` is sent to three messages. The messages are constantly updated with the three fader values. A `split` object ensures that if the fader value is within ± 0.06 of its previous

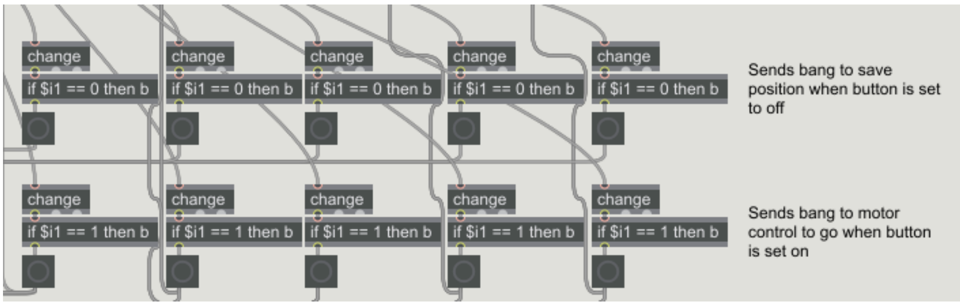

Figure 5.7: Effect selecting and deselecting filtering

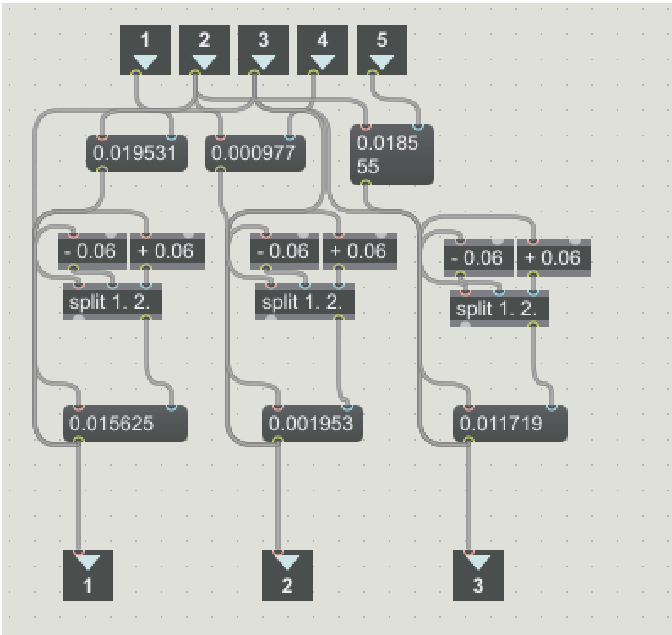

Figure 5.8: Previous fader location patch

value it will not overwrite the value. This is to make sure that any slight overshooting by the fader is ignored.

The motorised faders used in this project are DC motors. This means H-bridges are necessary to switch the direction of the motor. Directions are switched with a message to two pins (per H-bridge) with 0 and 1 for left or 1 and 0 for right. An if-statement controls the direction. In pseudo code:

```
if saved pos > current pos
  then right
    else
      left
```

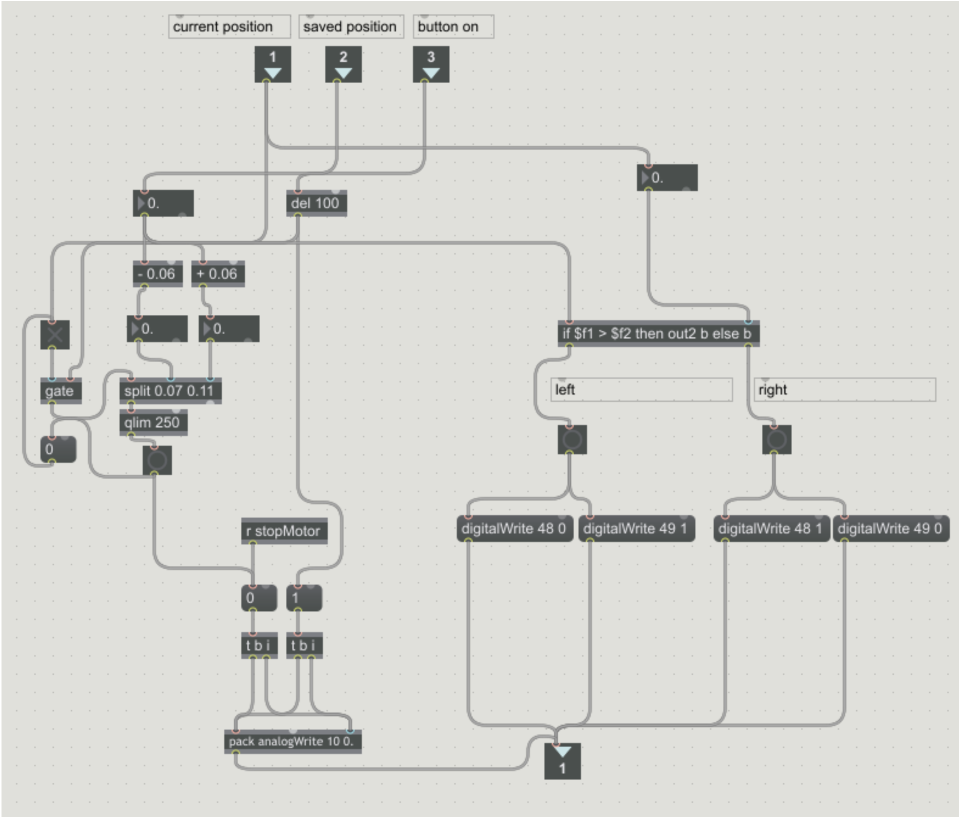

Figure 5.9: Fader control

This can be seen on the right side of figure 5.9. The function of the motorised faders is to resume the position a particular effect was on when selected or activated again. Trying to accomplish this with DC motors and potentiometers means: setting a target (i.e. the saved position); setting the correct direction; starting the motor; stopping the motor once the target has been reached and, finally, making sure that everything is set up to do it again. The main issue was overshooting the target. The motor speed was too fast and the potentiometer readings would skip numbers and thus the motor would never stop. This was solved by using the `split` object. The `split` object sends the number out of the left inlet if the input is within a certain number range. This number range is determined by using the saved position ± 0.06. To make sure no messages to the motor are sent once the target has been reached, a gate before the current position number flow is activated alongside the motor start. Gates open and close if they receive a 1 or 0 in the left inlet. The gate is closed once the target has been reached. A trigger object is used in the end to make sure the message being sent to the motor pin has the correct affix (0 for stop 1 for start).

Once an effect is selected or activated the fader moves to the target position. This

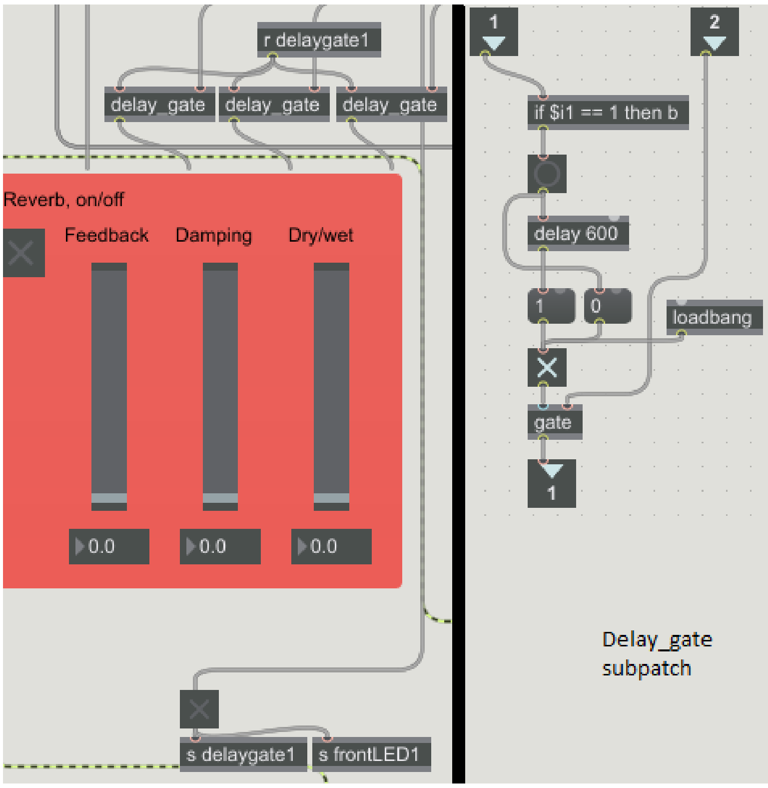

Figure 5.10: Delaying parameter change on activation

also means the parameters change with them. This can be problematic for many effects. The pitch shift could go through numerous pitches which would ruin the flow of music. Therefore, the fader values are gated behind the delay gate subpatch which closes the gate for 600 milliseconds when an effect is activated.

5.4 RGB LEDs

RGB LEDs are controlled by PWM. In terms of Max, it means sending a number between 0 and 1 to the correct pin. As can be seen in Fig. 5.11 the colours can also be combined. The effect with the highest amount of settings is delay, as such, six different colours are needed. This patch takes the fader position which routes to a certain colour.

Some effect have continuous parameters. This is true for reverb. A dry/wet mix set-up (see Fig. 5.12) is used to change from one colour to the next.

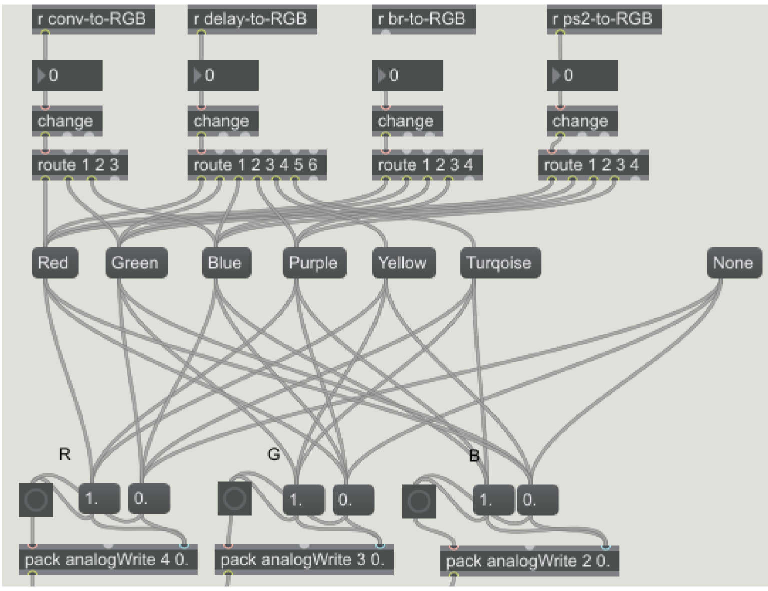

Figure 5.11: RGB LEDs controlled by fader position

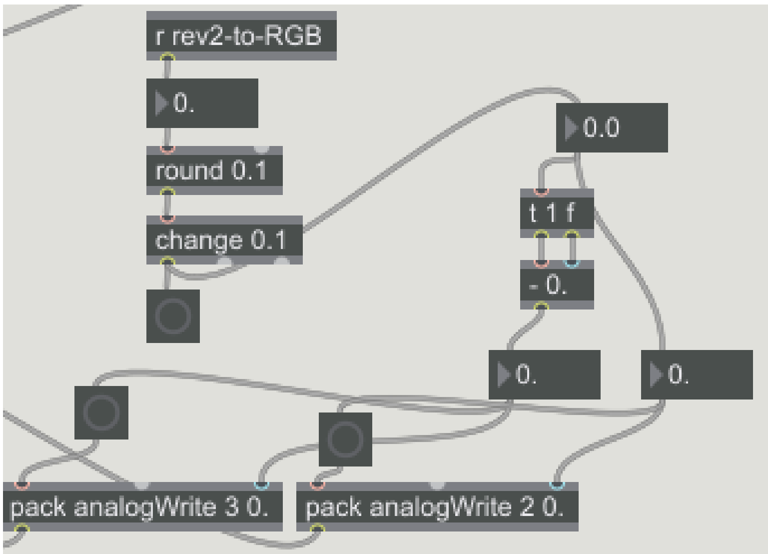

Figure 5.12: RGB LEDs analogue colour logic

Chapter 6

First Prototype

This chapter describes the first prototype created as well as its hardware and software elements. Tests were also conducted and are described here.

Naturally, we needed to create a prototype to test the ideas behind a hand-operated controller. The first iteration controller was a wooden box, large enough to house all electronics we were working with. Likewise, all the controls also fit into this one box for the sake of simplicity and to make it easier to build.

6.1 Hardware

Based on chapter 3, the box was designed to contained 10 simple push buttons and 3 motorised faders. The box can be seen in fig. 6.1. Short-circuit type buttons were chosen over toggle switches which have an inherent state-behaviour because it did not restrict the type of mapping that could be done in software. The red buttons on top are used to activate and deactivate effects. The black buttons on the front are used to select which effect is being manipulated. The three faders are used for changing the chosen effect's parameters. To ensure that the faders are usable for all effects, motorised faders were chosen in order to restore previous settings of effects when switching between them. The box also contained an Arduino Mega, which was used for reading the values/input from both buttons and faders as well as controlling the fader motors. Capacitive sliders and rotary encoders were also considered for parameter controls because of their stateless nature but were discarded because of their lack of visual feedback and although this could have been accomplished through LEDs we opted for the simpler solution for the first iteration. The first iteration also utilised a phone for its IMU capabilities. An application called TouchOSC was used to transmit IMU-data as OSC messages directly to Max/MSP [2].

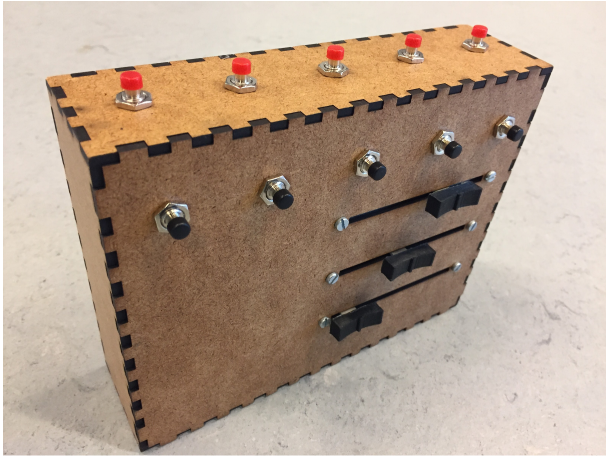

Figure 6.1: The control box for effects.

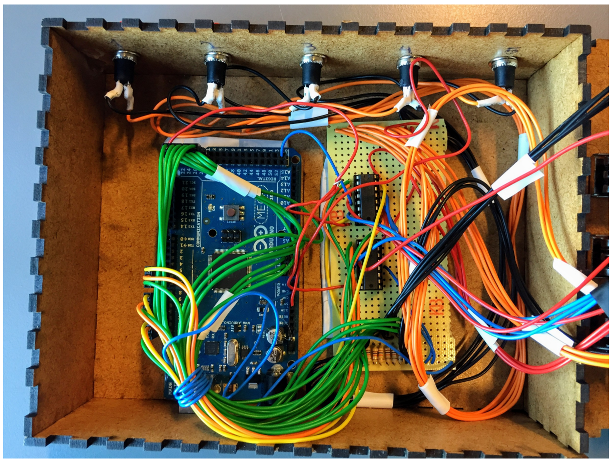

Figure 6.2: Inside the control box

6.2 Electronics

Even though all sensor data was read directly by the Arduino, we decided to gather all electronic connections into an intermediate board within the box and from there connect them to the Arduino as can be seen in fig. 6.2. all sensors were wired to the board where they were given VCC and GND as required. With the intermediate board, we were able to easily replace the Arduino and even replace it with a different processing unit if needed - the set-up is sketched in Fig. 6.3. The board also had a pair of L293Ds (H-bridges) that were used to control the fader motors via PWM signals.

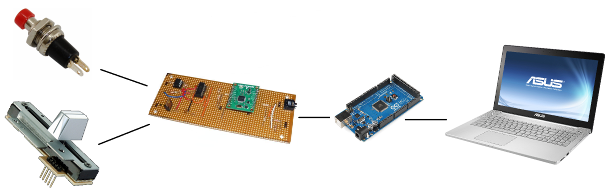

Figure 6.3: Overview of the electrical components and their connections.

6.3 The Box

The box was cut from a 3mm MDF board in the dimensions 180mm×140mm×50mm (length × width × height) using an Epilog 30 Fusion Laser Cutter & Engraver. The design was created using MakerCase [29] where the desired dimensions are input whereafter an SVG-file is created. We chose to do this because the box was easy to assemble due to the finger edge joints. The dimensions were based on the amount of space needed by the sensors, the electronics board, the Arduino and the wiring.

6.4 Software

All sensor data captured by the Arduino was sent to Max/MSP [2] using Firmata [30] code which essentially overwrites the OS of the Arduino and facilitates the Maxuino [28] software which we use within Max/MSP. The GUI available to the users can be seen in Fig. 6.4. The GUI showed the user which effects were activated, the parameters for each effect and also which effect was selected (i.e. which effect was currently being manipulated). It was also possible for the user to either set a BPM or tap in the desired tempo. It is important to note that the GUI is not going to be available to the performer when he is on stage. Rather, the GUI was to make it possible to do the usability test and it has the prospect of being used to fine tune effects for the user when not performing.

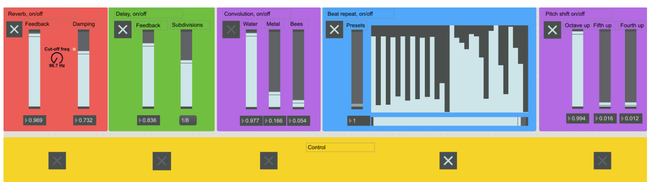

Figure 6.4: The Max/MSP GUI

6.5 Expert Test

The expert test was conducted in a rehearsal room at the royal academy of music in Aalborg. One person conducted the experiment with two participants: the collaborator and his double bass teacher. The participants were shown how the box functions after which they recorded a piece on a double bass and used it for the box. Each effect was used and commented upon, after which a semi-structured interview was conducted with the collaborator only. The different topics included strengths and weaknesses, how would the project change how you play, comments on sound effects and parameters, what hurdles remain for you to use the project effectively, comments on IMU, what genres does he currently play with a double bass, how can augmentation expand this.

6.5.1 Results

According to our collaborator, the double bass is locked in a few genres such as jazz, classical music and folk. He wants to challenge himself and the instrument he plays. The collaborator mentioned when discussing the different hurdles to overcome before being able to use the product effectively, that he would definitely need to practice with the instrument (because the augmentation makes it a different instrument altogether) and compose a music piece specifically for the augmented double bass. Therefore, as there were no outstanding problems with the design of the box, he could not comment in depth on it. However, many of the effects needed either new parameters or changes to their existing parameters such as dotted eights for the delay rate. Another problem identified was the IMU motion. Rather than bending forward or sideways it was deemed a much easier motion to simply turn. However, as the IMU was not essential to our collaborator, we decided not to pursue the matter further.

Notable changes to effects were dotted eights for delay rate. This was to enable an off beat, making a rhythm more complex and musically interesting. Depth on beat repeat effect to create more dynamically diverse options. The beat repeat effect was also previously outside the chain of effects, creating a background beat. However, this without any dry/wet slider to control the mix it would get overshadowed by all the other effects which made it less interesting for the test participants. Dry/wet mix for every effect and it should be on the same faders. This change is to streamline the effects, lessening the bandwidth required to memorise and use them in a performance. IMU could be used to control tempo on beat repeat and finally, pitch shift should have a dedicated fader for different octaves and fifths. The change to the pitch shift effect was to, again, streamline the effects with an added dry/wet mix faders. Ideally, a fourth fader could be added for more parameters.

6.5.2 Discussion

An important point to take from these results is how the collaborator feels about the double bass and how possible it is for the project to fix these problems. During the interview, we tried to establish what genres and playstyles he felt the bass was locked into, and it was mostly jazz and the occasional pop song with the same old bass line. This answers some of the original questions of the inherent role and capabilities of the double bass. The first initial idea to solve this problem seemed to be on the right path. This meant we could continue to the second iteration.

6.5.3 Second expert test

In a second smaller expert test, the device was mounted on the double bass. A pickup microphone through a soundcard connected to the computer provided sound. The collaborator tested the various effects again (done after changes to the effects were made from testing). The main points we took away from this test was the necessity

Figure 6.5: The mounted device.

of a more robust mounting system than simple elastic ropes for bicycles and adhesive putty (see Fig. 6.5). The surface of the double bass is uneven, so a mounting system would need to take this into account. A few solutions were considered such as a rig similar to that of a violin's neck brace. Another solution included using foam on the bottom of the box to fit the curved form of the instrument to a higher degree.

6.6 Usability test

A usability test was conducted to test the control box on its own where 7 students of the Sound and Music Computing master's programme at Aalborg University. participated in the test. The test consisted of 3 phases followed by a survey for the participants to provide feedback. The point of this test was also to deem whether or not the controls were good enough so that we could proceed with the expert test (see section 6.5). The test's tasks were is based on guidelines by the `usability.gov` website [31] while the questionnaire was based on a paper by Bin et al.[32]

6.6.1 Method

In the first phase, participants were introduced to the controls and a GUI (the same GUI that is seen in Fig. 6.4). After the introduction, they were then given a few moments to familiarise themselves with the controls and the GUI. In the second phase, the participants were asked to complete 10 tasks which included both activation/deactivation of effects, as well as changing parameters of the effects using the box. Throughout the tasks, a background track of a double bass was playing in order to provide audio feedback for the participants. In the final phase, participants were allowed to freely experiment with the controls and effects on the same background track from the tasks. After the test, the participants were asked to answer a survey where they had to rate a number of attributes on a scale of 1 to 7 along with optional comments on controls and effects. The rated attributes included: understanding of the box after each phase, understanding of effect parameters, satisfaction with the effects and overall satisfaction with the selection of effects.

6.6.2 Results

In the analysis of the results, we can see that there is a slight increase in understanding from phase 1 (introduction and quick familiarisation) to phase 2 (tasks). In phase 3 there is a slight and puzzling decrease (see Table 6.1). There could be a number of reasons to this. One being that the user had no clear goal laid out in front of them on a piece of paper. This might create situations where they do not know what exactly to do.

Table 6.1: Participant's understanding of controls

Understanding	Phase 1	Phase 2	Phase 3
	6.71	6.86	6.43

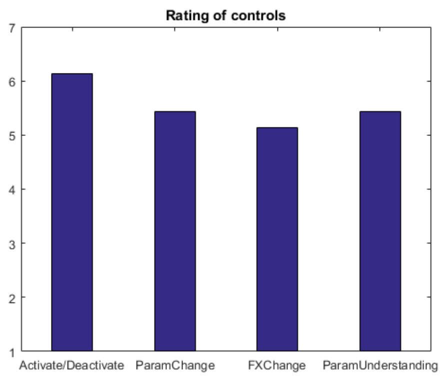

Figure 6.6: Ratings for Control

In Fig. (6.6) We see overall high scores. The figure shows how easy it was to activate and deactivate effects, change parameters, change which effect you are controlling and lastly how well the participants understood the parameters for the effects. The lowest score (FXChange) is understandable as it can be less intuitive to change which effect you are controlling but not actually turning anything on or off.

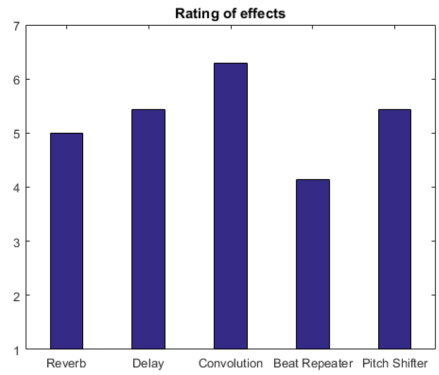

Figure 6.7: Ratings for sound effects

In Fig. (6.7) we see the scores for all effects. The more novel convolution effect received the highest scores while reverb and beat repeat lack behind. Based on the comments, these effects were changed. Reverb was altered so that smaller changes in the feedback parameter would have a greater effect. The cut-off frequency parameter was also changed to a dry/wet mix instead. The beat repeat effect was changed to affect 100% of the sound mix rather than the previous 50%.

6.6.3 Usability test discussion

The questionnaires were all set in the end of the test. They should have been immediately after their respective phase to avoid losing details. The questionnaire was based on a test made for audience members regarding their understanding and experience of a NIME concert [32]. While some results are gained, a more suited questionnaire such as the System Usability Scale (SUS) by Brooke [10] could be considered. The results are not without merit, however, and the test scores can be interpreted as a subjective score.

Chapter 7

Final Design

This chapter describes the second iteration prototype of the control units with its hardware and software components. This chapter also includes user tests of the prototype and discussion of the results.

7.1 Experiences from First Iteration

In general, our collaborator enjoyed the prototype. The controls were easy to use and responsive when activating and selecting effects. The faders worked well for both controlling the parameters of effect as well as serving as visual feedback to the settings of the parameters (e.g. when switching between effects). However, after testing some areas of improvement were identified within both hardware and software.

7.1.1 Hardware

First thing was the size of the box. Being fairly large, even compared to the double bass, it was difficult to place the box on the body of the instrument. This was especially true because the box was rotated 90°compared to the initial idea in order to fit. Secondly, the mounting method of combined elastic ropes and adhesive putty did not work very well. Thirdly, the user was too dependent on the laptop. The box at this point offered no feedback as to which effects were active and selected making audio cues the only alternative to consulting the screen. Furthermore, the BPM and the pattern for the beat repeat were only controllable within Max/MSP.

7.1.2 Software

A few areas of improvement were likewise found within the software. These areas mainly concern user experience with the box and changes to sound effects and how they are controlled. The first area was the changing faders. When the user switched between two active effects, the faders would, of course, move to match the settings of the effect changed to. But since the faders were read real-time changing to e.g. the delay effect could result in a change in the delay rate if the fader controlling delay rate had a travel distance before reaching the setting. Changing the delay rate causes the delay line to be reset which clears any current delays. The second area concerned the faders and their assigned parameters. The issue was that every effect had a completely different set-up concerning which faders affected what parameters. With reverb, the faders controlled, in the following order: feedback, cut-off frequency for lowpass filters, damping. For the delay the faders controlled only feedback and delay rate, leaving the last fader unassigned.

7.2 Design Choices

Based on the experiences with the first iteration a few changes needed to be made. The necessary changes and the choices behind them are described in the following subsections.

7.2.1 The Box

Two major issues were the size of the box and the mounting system. In order to reduce the size the box was split into two boxes: one containing all the sensors and one containing the Arduino itself (see figure 7.1). The idea was that the sensor box be mounted on the front of the instrument and the Arduino box be placed on the backside of the instrument where it is not in the way of the player. An option at this point would also be to make the connection between the Arduino and the laptop wireless, but we chose against this because a wired connection has no delay and because the double bass is quite the stationary instrument. This also made the process of prototyping easier. The

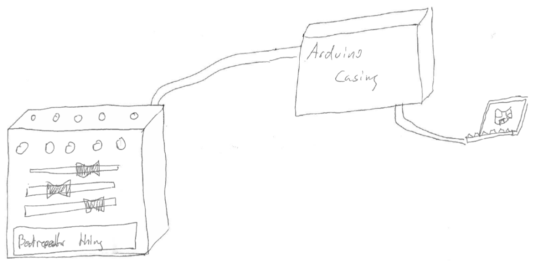

Figure 7.1: Sketch of to boxes, one with sensors one with Arduino. The latter being connected to the laptop

original box had the dimensions of 180×140×50mm which was too large. To determine the necessary size of the sensor box, the original box was modelled in SketchUp. Modelled versions of the buttons and faders were also created based on their actual dimensions (see fig. 7.2). Assuming these components, the smallest possible box had the dimensions 140×120×35mm. However, the small box was decided against due to

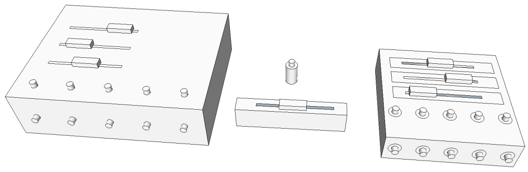

Figure 7.2: On the left is the model of the original box. In the middle, the models of a button and a fader can be seen. Finally, on the right is the model of the smallest possible box.

a fundamental design error. It could only just contain the sensors from the first iteration and space was needed for expansion in regards to adding more components such as LEDs for visual feedback. It also left small room for error and required a precise estimation and technique in terms of wiring which neither of the authors possessed. A slightly larger (160×140×40mm) box was thus modelled and can be seen in fig. 7.3 - this model was realised as the final iteration.

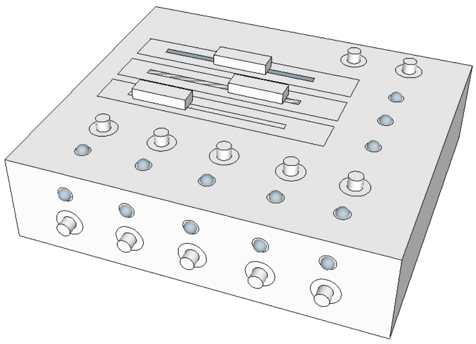

Figure 7.3: A model of the final box including LEDs.

See Fig. 7.4 for a picture of the final box.

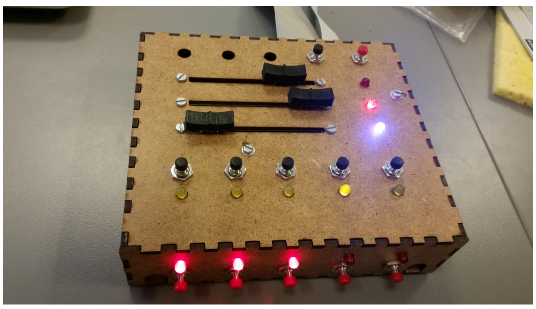

Figure 7.4: Picture of the final box design

7.2.2 Electronics

There were many changes to the hardware and thus also the electronics in the second iteration: a process that took several weeks. First of all, we added two more buttons and introduced 13 new sensors: 11 LEDs and 2 RGB LEDs. (The choice of adding LEDs is described in section 7.2.4.) Secondly, the initial box was split into two (one for the sensors and one for the Arduino) which meant that two boxes needed to be connected in some way.

In the first iteration, every component was connected to ground and +5V on the intermediate board via individual wiring. For the second iteration, a single wire for ground and likewise for +5V a supply were attached to each individual component. This can be seen in fig. 7.5 where each of the wires is marked. Each of the single wires was then connected to the intermediate board.

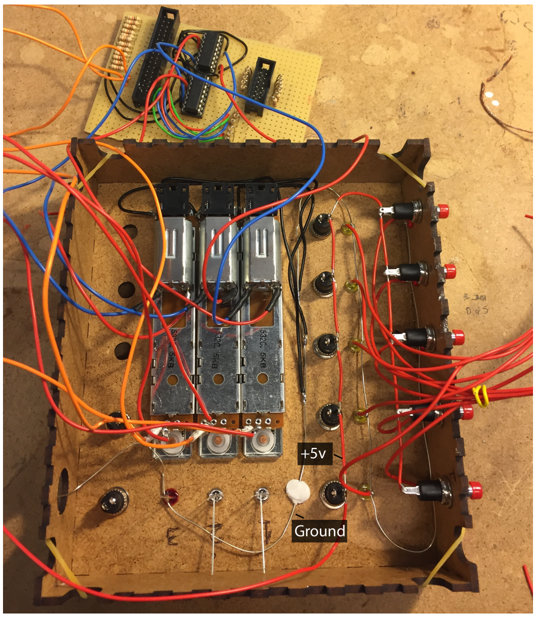

Figure 7.5: Inside the sensor box.

As in the first iteration, the intermediate board is where all connections were gathered before being connected to the Arduino. In the second iteration, however, the connections had to be made to the Arduino-box. Instead of using individual wires for the connection, we chose to use two ribbon cables with one end attached to the intermediate board.

The original idea was to make a similar intermediate board for the Arduino box and connect the ribbon cables to that. From there it would be possible to connect to the correct pins. Once again we were presented with a method that made the connection easier. There are many different kinds of shields for the Arduino (WiFi, RFID, BlueTooth etc.) which are customised to fit perfectly. With the custom shield, it

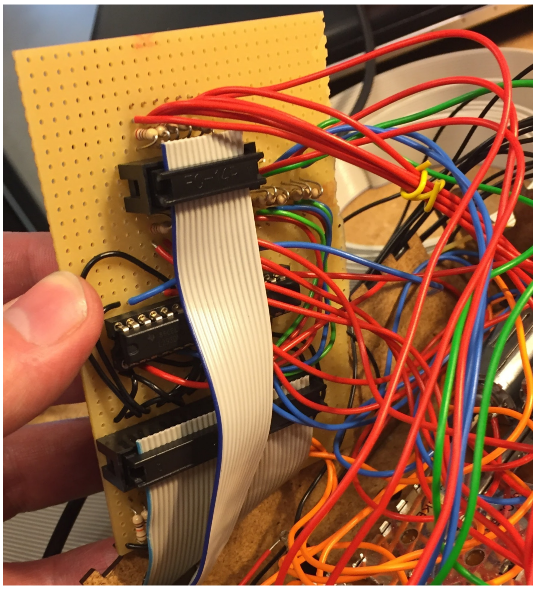

Figure 7.6: Inside the sensor box.

was possible to fit the ribbon cable housings directly on the shield and then connect the wires to the correct pins as can be seen in fig. 7.7.

As all pins on the Arduino can be programmed to be digital pins, sensors were at connected to pins based on convenience, i.e., the shortest distance from the cable housing to the pins (most of the Arduino's digital pins at the bottom which are seen on the right side of fig. 7.7). Unless, of course, the sensors required a specific type of pin such as the faders needing analogue input and motors plus RBG LEDs requiring PWM output. But issues arose when buttons were connected to analogue pins. This caused the value from the buttons to be fluctuating and toggling on and off randomly in the software. After this, all the sensors with on/off behaviour were wired to the digital pins.

7.2.3 A Mounting Method

The first iteration did not include an actual mounting system but for testing purposes, a combination of elastic ropes and adhesive putty were used for mounting the box on the bass. The main issue with this set-up is the incompatibility of the flat surface of the box and contours of the surface on the double bass which we attempted to counter by using the adhesive putty. But as it was not possible to place the elastic ropes in a way over the box that did not occlude controls and buttons, the adhesive putty was simply smeared over the surface of the bass as the box kept sliding out of place. The sliding happened because the elastic ropes only covered one corner of the box which resulted in the box being pulled askew. We did find that the elastic bands could be

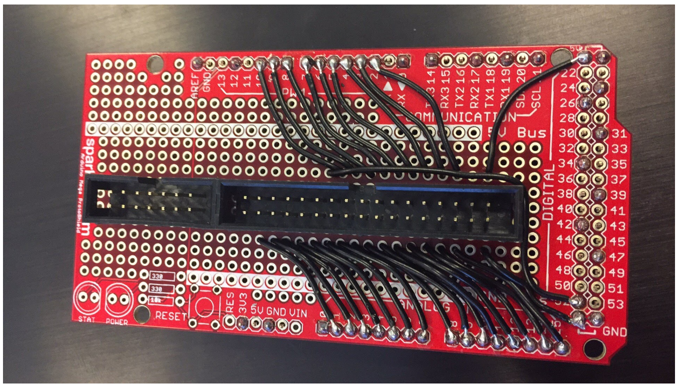

Figure 7.7: Shield for the Arduino with socket for the ribbon cable.

a used in a mounting system if appropriate changes were made to the box. E.g. by adding holes in the side of the box enabling the bands to pass through the inside of the box and thus ensure their hold on the box. Another consideration was the mismatch of surfaces. An idea could be to change the construct of the box to match the contours of the bass. A second idea could be to add some rubber "feet" to the backside of the box to create less area of contact. The final idea included a simple cut out of pieces of a regular washing sponge which was glued to the backside of each box (see figure 7.8 - the figure also shows the holes intended for the elastic ropes).

The above method was tested and found inadequate. There was no way to tie the elastic ropes to the bass and have the control box in the correct position without the box being pulled a skew (see fig. 7.9). This was a result of the elastic ropes being completely round and that their surface had no grip. A final idea for a mounting system used suction cups. Plastic fittings were 3d printed were attached to both boxes whereafter the suction cups were attached to the fittings (see Fig.7.10).

Not only was the result more aesthetically pleasing than previous iterations, it was also more robust and it was possible to get the box positioned correctly (see fig. 7.11). However, the system was not perfect. Because of the surface being made of wood and its inherent imperfections, the suction cups were not able to maintain a constant grip. To support the suction cups and to ensure that the box did not fall, a black elastic band (fashioned from a bike tire inner tube) was wrapped around the body of the bass and fastened with a 3d-printed buckle (see fig. 7.12).

7.2.4 Effect Parameter Feedback

The box of the first iteration had a simple build only containing the sensors needed for controls: i.e. buttons and faders. This also meant that the only feedback the user was given from the box was the positions of the faders which corresponded to a setting of parameters within Max/MSP. To know which audio effects were active the user would either have to listen for audio cues or consult the screen of the laptop. Ideally, the user

Figure 7.8: The sensor box with sponge attached.

should be able to tell both which effects are active as well as have an indication of their settings just by looking at the box.

To facilitate more feedback LEDs were considered. By placing a standard on/off LEDs near the activation and select buttons it would be able to indicate which effects were active and which one was currently selected. Two types of LEDs were also considered for an added feedback of the effect settings. By using a 10-segment light bar (which is essentially 10 on/off LEDs ordered in a row) it would be possible to give an additional feedback on effect settings (e.g. by turning on all LEDs with the maximum setting and vice versa with minimum setting). This idea was, however, discarded as there were not enough pins. It would potentially be possible with the remaining pins to use bit-shifter register but time constraints did not allow for this. Another option for a visual representation of parameter was to use RGB LEDs. They require one PWM pin per colour and by using indicative colour schemes we would be able to denote both a sliding value such as 0-1 feedback or discrete values such as 1, 2, or 3 for selecting pre-sets.

Our collaborator asked for an option to control the pattern of the beat repeat effect. We considered implementing a series of 16 buttons which would enable the user to control 16 parts of the beat repeat effect pattern (in pairs of two 16th notes). However, buttons would enable the user to set the value to 1 or 0 at the different time instances. Another consideration that would give the user more control was a series of FSR

Figure 7.9: The sensor box with sponge attached.

Figure 7.10: 3d printed fitting with suction cup attached.

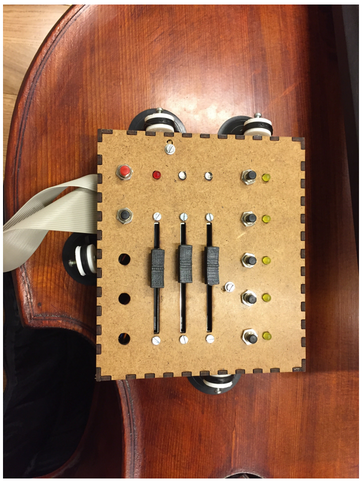

Figure 7.11: Box mounted on the double bass using suction cups.

Figure 7.12: Elastic band with 3d-printed buckle.

sliders which would give the user full control. In stead, we chose to use a smartphone with TouchOSC and created an interface for this, where the user could control all the 32 16th notes. The implementation and general details about the beat repeat effect can be found in section 4.6)

7.3 Testing

The project was tested and evaluated through a usability test and an expert test. Both are described in this section.

7.3.1 Usability Test (Second Iteration)

The second usability test uses the same tasks as the first iteration. However, it used the System Usability Scale (SUS) [10] to evaluate the box rather than the same method used in the first iteration usability test. Reasons are discussed in section 7.3.1.3.

7.3.1.1 Method

As mentioned above, a SUS is used to evaluate the system. It is a likert scale with 10 questions with 5 response options.

- I think that I would like to use this system frequently
- I found the system unnecessarily complex
- I thought the system was easy to use
- I think that I would need the support of a technical person to be able to use this system
- I found the various functions in this system were well integrated
- I thought there was too much inconsistency in this system
- I would imagine that most people would learn to use this system very quickly
- I found the system very cumbersome to use
- I felt very confident using the system
- I needed to learn a lot of things before I could get going with this system

For odd numbered questions you subtract 1 from the user response. For even numbered questions you subtract the user responses from 5. All values are then added up and multiplied by 2.5 which converts the values to 0 to 100. A score of 68-70 or above is above average [33][34]. This gives a short usability scale that can easily be given to the participants. It is also robust and can be used with a small sample size with valid results.

The test was conducted on 11 different people between the age of 23 to 27. All participants were students at Aalborg university with at least a B.Sc.
The box was mounted on a cello that was put at an appropriate height to mimic the feel of a double bass.

An introduction to the project and commencing test was read aloud to the participant (see appendix E.1) before they were walked through the controls of the box. They were then put through 11 tasks (see appendix E.2) that walks them through the capabilities of the box and functions. They then filled out a questionnaire followed by the chance to try the box using the same backing track they used for the tasks, or they could play the cello, also using the box.

A pilot test was conducted before the experiment.

7.3.1.2 Results

After processing the results as described by Brooke [10] we reach a mean score of 73.18.

7.3.1.3 Conclusion and Discussion

We can conclude that the usability factor is above average and thus has the necessary ease of use that is wanted from a product. The most common criticism that was made was the direction of the faders. The faders minimum is at the top of the box and maximum at the bottom. This was a consequence of having the box rotate 90 degrees between first and second iteration. On reflection, it makes sense that someone who had not operated the box before would make this observation. An argument for keeping this control scheme is, at that the bass player moves his hand downwards on the strings to increase pitch and it made sense to transfer the same mechanic to effect parameter controls. But as the test participants were only testing the box, this reference was not available to them.

There was also the issue of having to look at the box constantly. While we had hoped to provide sufficient feedback, it takes time to get familiar with the controls. More time than participants had during a usability test. All participants expressed their enthusiasm for the motorised faders and the colourful LEDs.

All participants were young students of technologically strong fields. Participants of this nature will typically have an easier time adapting to new technology. This test might receive different results if tested on a wider range of users. However, as the target group for this project is aimed at experienced musicians who are already interested in effects and expanding their capabilities through technology, we deemed this an acceptable variable.

7.3.2 Expert Test

At the end of the project, we interviewed our collaborator in order to review the results of the project. The questions, answers and comments for this interview can be found in appendix D. Throughout the interview, our collaborator referred to our project as "the box".

At this point, our collaborator had had the box for approximately a month. When asked about how he found it, he was very positive saying that he enjoyed it and that it really gave a lot of musical options. Words such as "cool" and "awesome" were frequently used. But there was also a comment about the entire set-up which, at this point, required too much gear, which made it a very cumbersome task. However, when it was up and running, it was a pleasure to use, earning admiration from fellow musicians.

Through the testing period, our collaborator had used the box and during this time it was already used for a live performance and planned to use it again in a second performance planned. The completed performance was called "Bag Masken - Digte om Angst" (Behind the Mask, poems about anxiety) and the planned performance was a theatre piece.

During Bag Masken (see Fig. 7.13), our collaborator played the double bass and a synthesizer alongside a guitarist, a trumpeter and an electronic musician while poems were read aloud. Our collaborator used reverb, convolution and pitch shifting in his performance. Unfortunately, the mounting system was not ready at this time and he placed it next to him on a table. He really enjoyed using the project for this performance. It was especially entertaining to play alongside an electronic musician because at some moments it was hard to tell exactly who was doing what. Despite the fun, he once again commented that there were a lot of technical issues before with setting up the project which led to a delay in starting the performance.

Figure 7.13: Bag Masken performance at Aalborg Hovedbibliotek.

In the last expert test, it was established that the role of the double bass is typically part of the rhythm section, supporting a band from the background. But with our project, our collaborator definitely felt, that his role had been extended. He was now able to really step out of the background and play a more soloistic role.

Only five audio effects were available at this point. The collaborator commented that it was an adequate number of effects when asked about it, at the same time mentioning that limitations are great for sparking creativity. The current selection had enough range to create anything from subtle effects all the way to massive and noisy soundscapes. By keeping the selection manageable he felt that the box had "personality". He did mention that perhaps it would be useful to have pre-sets of effects e.g. customise the selection for a specific genre (pop, jazz, electronic etc.). The number of effects would still be limited to five but with a large library of effects, he would really be able to tailor the box to any genre and performance.

We also asked him if there were any changes or additions, he deemed necessary for the box. Once again, he told as that the box was exactly as complex as it needed to be. While the IMU would have been interesting to experiment with, it was not needed at this point. Any or all expansions would mostly be interesting if he used the box in a solo performance but at this time he solely used it in group performances.

In the beginning of the project, both the collaborator and we agreed on the idea that the double bass was mostly found in jazz, classic and folk/country music. The "Bag Masken" performance was mostly improvisational jazz but the theatre piece was actually electronic pop. It was easy for our collaborator to imagine going further and deeper into the electronic genre. Our project made the bass sound very "produced/processed" and electronic - especially considering that the double bass is purely an acoustic instrument. But there was basically no genre in which it could not be used.

When asked about any hurdles or obstacles with the project, he once again pointed out that setting up the box was the major issue. It could easily take up to 30 minutes to set up all the necessary gear (i.e. a laptop, an external soundcard, a stand for the laptop and soundcard and cables). Because of this, he admitted to only using the box when there was a rehearsal. If the task of setting up was easier, he would use it all the time.

In this case, it is interesting to compare our collaborator's set-up to that of the electronic musician in both performances. The electronic musician had an extensive set-up also consisting of a laptop and a soundcard as well as three large USB-controllers. Although these were commercial products, perhaps it is just a matter of getting used to the extra set-up since our collaborator usually only had to un-bag his bass, plug in a jack cable and he is ready to play.

Final comments about the project mostly included praise and exclamations of satisfaction with the results. Our collaborator was very interested in furthering the project to fix issues and, hopefully, bring it to an easier-to-set-up state such as an embedded platform.

7.3.2.1 Summary

Overall, our collaborator was pleased with the results. He enjoyed playing the bass using our project and found it a lot of fun to use. There were no limits as to which genres, he could play, and at this point, he was already using our project for live performances. The controls were easy to use and the selection of effects was just right. On the down-side, he told us that it was cumbersome and time-consuming to set up. A lot of gear is needed and because of this he only used our project during rehearsals.

Chapter 8

Discussion

We set out from the start to augment the double bass. To give it new modes of interaction. It is arguable whether or not it is an actual augmented double bass. While it is certain that our collaborators' wish to move beyond the normal possibilities and roles for a double bass has been fulfilled, it is less certain if our solution is not just a control box for effects. The Augmented Instruments Lab in the Queen Mary University of London defines it as "An augmented instrument is a traditional musical instrument whose capabilities have been electronically extended through new sensors, new types of sound production or new modes of interaction" [35].

Considering this definition, without the IMU capabilities that have not yet been implemented, there are no new sensors except for buttons and faders directly on the box itself, the sound production is the same as before only with a microphone. New modes of interaction, however, is debatable. Using various effects, especially convolution and pitch shift, it is possible to make completely new sounds that are specific to this set-up of effects and choice of parameters. For example, hitting the body of a normal double bass can be used to create a beat but it does not go much beyond that hit. A dull thud depending on the region of the bass you use. If used in combination with convolution and delay, for example, entire soundscapes are made possible by just one performer and on the fly switch back to normal double bass playing. This is especially true when using the pitch shift effect which can make the double bass sound like an entire string section or even a synthesiser. In this way, new modes of interacting with the bass are opened up.

Mounting system notwithstanding, the system is also instrument agnostic in the sense that any instrument you can microphone up can use it. Our collaborator makes use of this by combining it with his synthesiser. This shows us that porting this to another instrument would be quite easy. The main issue with this is mounting. In the case of the synthesiser, he simply had it next to him on the table. The size of the box makes it difficult to mount on anything else than the double bass. Making a dedicated port to another instrument would require a redesign of the box. A different observation related to the mounting system was how it dampened the instrument. Two boxes are strapped onto the instrument directly using a large elastic band. This naturally dampens the sound output as it prohibits the body of the bass from vibrating. This was, however, not an issue when using effects.

During the project, we had been asked to advise on a project by students from art and technology. They wished to employ techniques we had studied during our time on Sound and Music Computing. This prompted the question: Would not it be interesting to have cross-studies meetings to have a wide range of opinions on how to solve project problems. In our case, the project ranges from both industrial design for the physical box building and mounting system and design work for user interface.

In testing the product, the plan was to test both iterations against each other. This was not feasible as the test was changed after reflection. As was mentioned in section 6.6.3, the test was built on a test that had a different context. Namely people's understanding of a NIME after watching a performance. The metric of understanding can be useful but it was perhaps built on the wrong foundation. As a result, we wanted to find a more robust test that could stand on its own without us having to compare it to the previous test in the first iteration. This led to the use of the SUS test. A helpful tool to make sure the user interface is intuitive and easily usable. The test resulted in a mean score of 73.18 where anything about 68-70 is considered above average. We can conclude that the user interface was easy to understand and use.

The real challenge was to evaluate it as a tool for musicians, if not an instrument. Having our collaborator test it over a longer period was very helpful and gave a lot of useful feedback, but as the case is for many tools and instruments you can continue to work and make new iterations for a long time.

This project was an interesting experience in that we worked with both a consumer and a collaborator. This provided us with a clear goal: a finished product that our collaborator could utilise in a professional setting. But it also gave us a few challenges. This becomes apparent when the research side of the project goes in the background for a bit until we remember it again.

Chapter 9

Conclusion

The goal of our project was to expand the possibilities and roles of the double bass and to answer the question: "How can we provide tools and possibilities for the double bass to expand its inherent roles and capabilities already found in the instrument?" with the sub-goals of "What is the inherent role of the double bass?", "What are its capabilities?" and "Does our tool change either of these?".

Through interviews with the collaborator, the role and capabilities of the double bass were defined. He feels it is locked into jazz, symphony orchestra and the occasional pop song. And only as part of the rhythm group, doing the same figures over and over. The solution of adding audio effects that can complement and expand on the double bass has given him new ways of expressing himself. Especially creating soundscapes is easily achievable.

The goal of expanding the possibilities and roles for the double bass was achieved through our project, thus making the project a success. Although the system we created for our collaborator still has a few issue such as the mounting system and the cumbersome task of setting it up, we succeeded in creating a product that is usable for our collaborator. The system allows the user to control, in real-time, audio effects and their various parameters and this was used for multiple live performances. This was especially true for a theatre performance where our collaborator used the system. He switched between playing the traditional role a bass player (i.e. playing root notes in chords) to using the bass to create accompanying and immersive soundscapes using the audio effects and the ability to change the parameters on the fly.

Even though the system has been used for live performances, it still requires a bit of work before it can be called finished. In its current state, it is difficult to change anything about the time it takes to set up without moving from a computer to an embedded platform. The mounting system, however, can be improved. At the moment it stays relatively safe on the double bass but it is cumbersome to move and taking it off. In the end, through the use of audio effects and giving easy access of these to the user, a double bass player can move beyond their traditional role and move into new ways of expressing themselves.

Chapter 10

Future Work

Although, our collaborator was very pleased with the results of the project and had already used the set-up for several live performances, there are many possible iterations do be done on this project, had there been more time.

Fine tuning the effects and parameters alone are worthy of multiple iterations. This could also include new and more novel effects can also be considered to further expand the role of the double bass such as rhythmic processing.

The controller set-up of the project is another area to go into. The sensor box stands out a lot from the bass because of its form and edges. Creating a controller that fits the unevenness of the bass would not only alleviate some of the mounting issues but it might also push to controller towards a more professional and finished look. The design mentioned in section 2.1.5 is especially interesting as a flexible material would be perfect for the curvature of the double bass.

As for the controls, it would be interesting to explore new methods of controlling effects and parameters. Further research could be done in the IMU but also completely different controllers including anything from motion or eye-tracking to galvanic skin response etc. Any kind of controller that does not require the hands to be operated.

One of the bigger changes would be moving towards an embedded platform. At this point, quite a lot of time goes into setting up the entire system. While steps have been taken to ensure that the program loads everything necessary on start-up, you still need to make sure that everything is correctly connected before booting up Max. This includes making sure you are using the correct audio drivers, that the speakers are set up and so on. This is also true for the sound processing because Max has to share the system resources with the operating system as well as any other running software.

An interesting addition to the project would be audio effect packs. Essentially, a pack would be five different effects that work particularly well together. As it is, the code is set up so that you can easily bring in new effects. This could be expanded to entire packs. This could be a way to bring new products to a user continuously. Interestingly, during the second expert interview, a point came up. The box has its own sound or "feel" to it. During the longer testing period, the collaborator gained an understanding of these particular effects and how he could quickly manipulate these. Changing the effects chain is also easy but a dedicated system for the user could be an interesting addition as well.

Another consideration is the possible expansion to other instruments. The biggest problem is the mounting system. As it is, the double bass gave us a lot of space to work with, which was helpful in prototyping as size was less of an issue. Nothing else stops you from plugging something else through, in fact, our collaborator used his synthesizer with it.

This project is closely related to the NIME conference and a new paper with the updates that happened during the second iteration can be made.

With some work, the Bass Augmentation and Enhancement System could be updated to a commercial product. When describing the project to especially musicians, there was a lot of interest in the idea of being able to easily add effects with a custom-built controller. A crucial area is optimisation of the manufacturing process. A lot of time went into connecting the electrical components to the intermediate board and from there to the Arduino. Ideally, a print board would be made containing all the components and connections, after which a box or casing is simply slid over. A consequence of this could be a controller box that was smaller, and with a more flexible mounting system, this would enable the use across other acoustic string instruments, e.g., the violin, cello or guitar etc.. If there was no need for mounting the controller, any instrument could be used with our system as long as the sound is captured with a microphone.

Bibliography

[1] Dan Newton and M.T. Marshall. "The augmentalist: enabling musicians to develop augmented musical instruments". In: *Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction*. ACM, 2011, pp. 249–252. URL: `http://portal.acm.org/citation.cfm?id=1935751`.

[2] Cycling'74. *Max*. URL: `https://cycling74.com/products/max/{\#}.Vx3ZxUx96Uk` (visited on 04/25/2016).

[3] Dan Overholt and Steven Gelineck. "Design & Evaluation of an Accessible Hybrid Violin Platform". In: *Proceedings of the International Conference on New Interfaces for Musical Expression* (2014), pp. 122–125. URL: `http://www.nime.org/proceedings/2014/nime2014{\_}470.pdf`.

[4] Peter Dahlstedt. "Mapping Strategies and Sound Engine Design for an Augmented Hybrid Piano". In: *Proceedings of the International Conference on New Interfaces for Musical Expression* (2015), pp. 271–276.

[5] Sébastien Schiesser and Jan C Schacher. "SABRe : The Augmented Bass Clarinet". In: *NIME 2012 Proceedings of the International Conference on New Interfaces for Musical Expression* (2012), pp. 109–112.

[6] .

[7] Bill Verplank. "Interaction design sketchbook". In: *Unpublished paper for CCRMA course Music 250a* (2003), pp. 1–23. ISSN: 07479360. DOI: `10.2307/1511669`.

[8] Sarah Nicolls. "Seeking Out the Spaces Between: Using Improvisation in Collaborative Composition with Interactive Technology". In: *Leonardo Music Journal* 20 (2010), pp. 47–55.

[9] Sile O'Modhrain. "A Framework for the Evaluation of Digital Musical Instruments". In: *Computer Music Journal* 35.1 (2011), pp. 28–42. ISSN: 0148-9267. DOI: `10.1162/COMJ_a_00038`.

[10] John Brooke. "SUS-A quick and dirty usability scale". In: *Usability evaluation in industry* 189.194 (1996), pp. 4–7.

[11] Udo Zölzer. *DAFX: Digital Audio Effects: Second Edition*. Vol. 4. 2011, pp. 0–471. ISBN: 9780470665992. DOI: `10.1002/9781119991298`. arXiv: `arXiv:1011.1669v3`.

[12] Keith Lent. "An Efficient Method for Sampled Sounds Pitch Shifting Digitally". In: 13.4 (1989), pp. 65–71.

[13] Richard Dudas. *Spectral Envelope Correction for Real-Time Transposition*. 2002.

[14] Emmanuel Vincent. *MUSHRAM - Matlab Interface for MUSHRA listening T*. 2005. URL: `http://c4dm.eecs.qmul.ac.uk/downloads/{\#}mushram` (visited on 03/29/2017).

[15] International Telecommunication Union. "ITU-R BS.1534-3, Method for the subjective assessment of intermediate quality level of audio systems". In: *ITU-R Recommendation* 1534-3 (2015).

[16] Groove Monkey. *Bass Freebie Pak – Groove Monkee*. URL: `https://groovemonkee.com/products/free-bass-pak-1` (visited on 03/29/2017).

[17] Karoryfer. *Meatbass - Karoryfer Samples - Karoryfer Lecolds*. 2015. URL: `http://www.karoryfer.com/karoryfer-samples/wydawnictwa/meatbass` (visited on 03/27/2017).

[18] Minitab. "Types of t-tests". In: (). URL: `http://support.minitab.com/en-us/minitab/17/topic-library/basic-statistics-and-graphs/hypothesis-tests/tests-of-means/types-of-t-tests/`.

[19] Pierre Alexandre, Alexander Harker, and Pierre Alexandre Tremblay. "University of Huddersfield Repository Original Citation Toolbox : Convolution for the Masses . In : ICMC 2012 : Non-cochlear Sound . The International This version is available at http://eprints.hud.ac.uk/14897/ The University Repository is a digital coll". In: (2012).

[20] Sophocles J. Orfanidis. *Introduction to Signal Processing*. Pearson Education, Inc., 2010. ISBN: 0-13-209172-0. URL: `http://www.ece.rutgers.edu/{~}orfanidi/i2sp`.

[21] Manfred R. Schroeder. "Natural Sounding Artificial Reverberation". In: *Journal of the Audio Engineering Society* 10.3 (1962), pp. 219–223.

[22] Mitchell Turner. *Mitchell Turner*. URL: `http://home.lagrange.edu/mturner/MitchWebSite/home.html` (visited on 04/25/2016).

[23] Hexler. *h e x l e r . n e t | TouchOSC*. URL: `https://hexler.net/software/touchosc` (visited on 04/19/2017).

[24] *Diego Stocco*. URL: `http://www.diegostocco.com/` (visited on 03/07/2016).

[25] David Gage. *Realist Copperhead or WoodTone for Bass*. URL: `http://www.davidgage.com/store/product{\_}info.php?cPath=21{\&}products{\_}id=28{\&}osCsid=4fb1b609cb0b236f585f2a2eda01e139` (visited on 05/19/2017).

[26] Gino Sigismondi, Rick Waller, and Tim Vear. *RECORDING MICROPHONE TECHNIQUES 3 RECORDING Microphone Techniques for*. 2014. URL: http://cdn.shure.com/publication/upload/837/microphone{_}techniques{_}for{_}recording{_}english.pdf.

[27] David Miles Huber and Robert E. Runstein. *Modern Recording Techniques*. Focal Press, 2009, p. 673. ISBN: 0240810694, 9780240810690. DOI: 10.2307/3679898. arXiv: arXiv:1011.1669v3. URL: http://books.google.com/books?id=W9U7A-rSXtEC.

[28] Maxuino. *Maxuino http://www.maxuino.org/*. 2010. URL: http://www.maxuino.org/ (visited on 12/06/2016).

[29] Jon Hallander. *MakerCase - Easy Laser Cut Case Design, http://www.makercase.com/*. 2014. URL: http://www.makercase.com/ (visited on 12/06/2016).

[30] Hans-christoph Steiner. "Firmata: Towards making microcontrollers act like extensions of the computer". In: *New Interfaces for Musical Expression* (2009), pp. 125–130. URL: http://archive.notam02.no/arkiv/proceedings/NIME2009/nime2009/pdf/author/nm090182.pdf.

[31] Usability.gov. *Running a Usability Test*. 2014. URL: https://www.usability.gov/how-to-and-tools/methods/running-usability-tests.html (visited on 04/18/2017).

[32] S Astrid Bin, Nick Bryan-Kinns, and Andrew P Mcpherson. "Skip the Pre-Concert Demo: How Technical Familiarity and Musical Style Affect Audience Response". In: *NIME '16 Proceedings of the 2016 Conference on New Interfaces for Musical Expression* (2016), pp. 200–205.

[33] Jeff Sauro. *MeasuringU: Measuring Usability with the System Usability Scale (SUS)*. URL: https://measuringu.com/sus/ (visited on 04/18/2017).

[34] Aaron Bangor, Philip Kortum, and James Miller. "Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale". In: *Journal of Usability Studies* 4.3 (2009), pp. 114–123. URL: http://www.usabilityprofessionals.org..

[35] Andrew McPherson. *Augmented Instruments Laboratory, C4DM*. 2017. URL: http://www.eecs.qmul.ac.uk/{~}andrewm/index.html (visited on 05/03/2017).

Appendix A

Electronics Design

This appendix covers some of the implementations of electronics in more detail.

A.1 Iteration One

The electronic setup of iteration 1 was fairly simple consisting of short-circuit buttons, motorised faders and a pair of ICs.

A.1.1 Buttons

10 short-circuit buttons (B163B) were each connected to +5V and a digital pin on the Arduino. However, because of the high-impedance state nature of the Arduino's input pins, digital pins switch between 0-1 with very little current which in turn makes input pins susceptible to noise. To combat fluctuations, the pin-connection of each button was connected to ground via a 10 kOhm resistor.

A.1.2 Faders

The three motorised faders (RS60N11M9_5KB) are basically slide potentiometers, were all connected to both ground and +5v through their ground and input terminals respectively. The output terminal of each fader was connected to a analogue pin on the Arduino. The motors attached to each fader were DC motors meaning that depending on which terminals were connected to ground and supply voltage they would only spin one way, so we used an H-bridge (L293D) to be able to change the direction of the current. Each motor's terminals were connected to the driver output pins of the IC. The enable pins were connected to PWM-pins on the Arduino, and the driver inputs (controlling the direction) were connected to digital outputs on the Arduino. Lastly, the IC needed both +5V and ground to function as well as supply voltage for the motors. Following the specification of the motors, they needed +10-12V to run, but it was

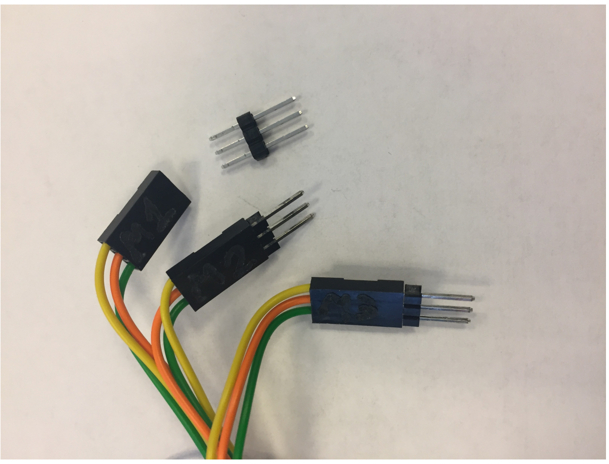

Figure A.1: Wires fitted with housings and pin headers.

possible to make them run on as little as +5V which mean that we could simply use the +5V output from the Arduino.

A.1.3 Intermediate Board

Only the faders' output terminals could be connected directly to the Arduino; the buttons all needed pull-down resistors and we needed to place the ICs somewhere. Therefore, we chose to have all connections run via an intermediate board (the right side of figure A.2 which also enabled us to gather all connections conveniently into housings. When fitted with pin headers (see fig. A.1) the result was a robust and easy way of connecting the sensors to the Arduino as well as disconnecting them.

A complete schematic of the electrical connections can be seen in fig. A.3.

A.2 Iteration Two

Several additions and changes were made to the electronics in iteration 2: Two more buttons and several LEDs (both regular and RGB) were added. The buttons and faders plus their motors were connected in the same manner as in iteration 1.

A.2.1 LEDs

Iteration 2 featured two types of LEDs. 11 regular LEDs were used: 6 red LEDs (L53SYD) and 5 yellow LEDs (L53SRC). The LEDs' cathodes were all connected to ground and their anodes to a digital pin on the Arduino through a 100 Ohm resistor to reduce the light intensity. Two common cathode RGB LEDs (LL-509RGBC2E-006) were also used. RGB LEDs of this type have 4 pins, one for ground and one for each

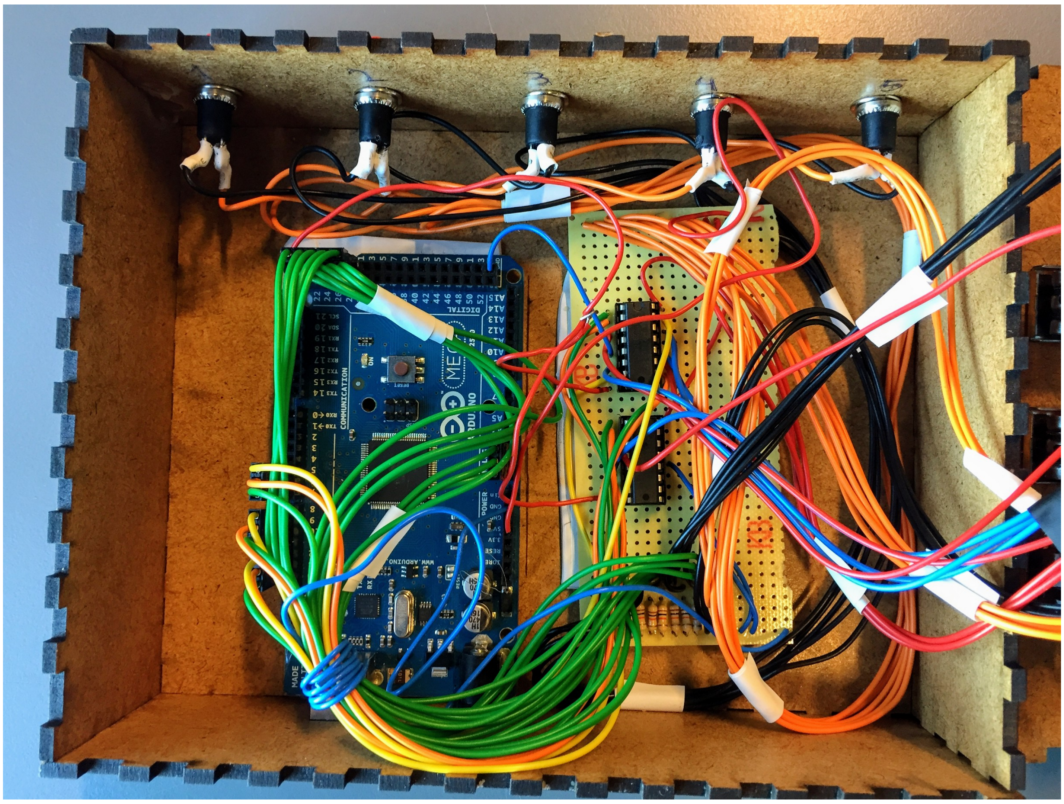

Figure A.2: Inside the control box. On the right is the intermediate board where all connections are gathered before connected to the Arduino.

colour i.e. red, green and blue. Each colour-pin is connected to a PWM pin through a resistor. Regular digital pins would have worked as well but with PWM it is possible to control the amount of each colour which enables detailed colour control.

A.3 Intermediate Board and Arduino Shield

In the second iteration the original box was replaced by two smaller boxes: one to house the sensors and one to house the Arduino. The sensor box still contained an intermediate board to which the sensors were directly connected. From the intermediate board all connections were now gathered into two ribbon cables which ran between the sensor box and the Arduino box (see fig. A.4).

A complete schematic of the electrical connections can be seen in fig. A.6.

In stead of connecting the other end the cables directly to the Arduino they was attached to an Arduino shield (DEV-09346) which can be seen in fig. A.5. Once again a robust and easy way of connecting sensors to the Arduino was achieved.

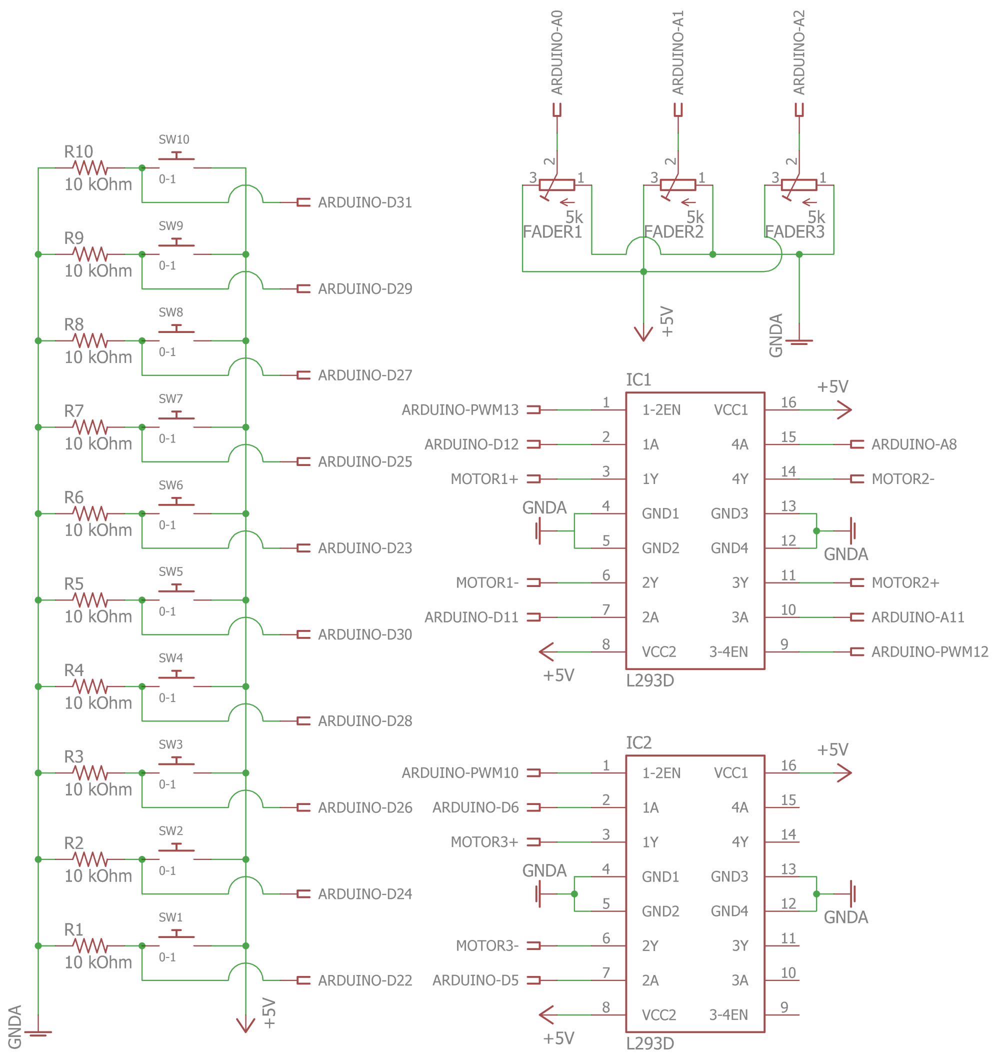

Figure A.3: The schematic of the electrical components as well as their connections.

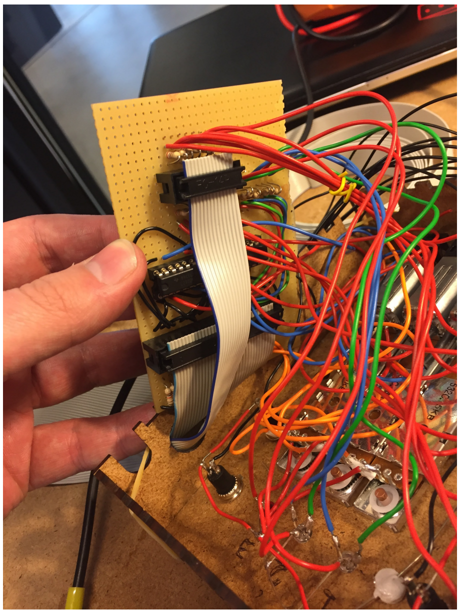

Figure A.4: Intermediate board with ribbon cables.

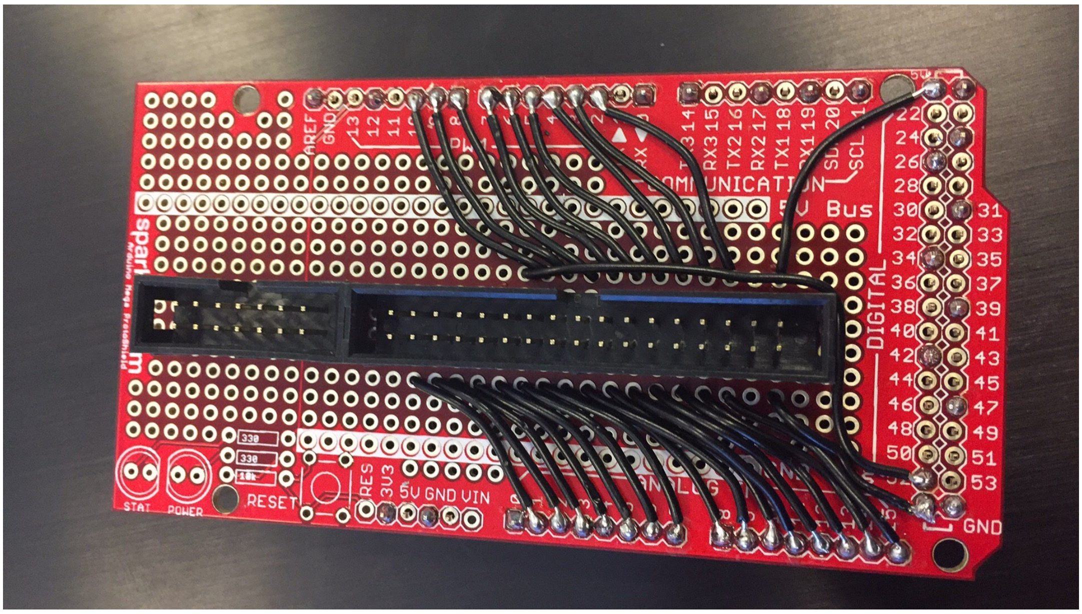

Figure A.5: Shield for the Arduino with socket for the ribbon cable.

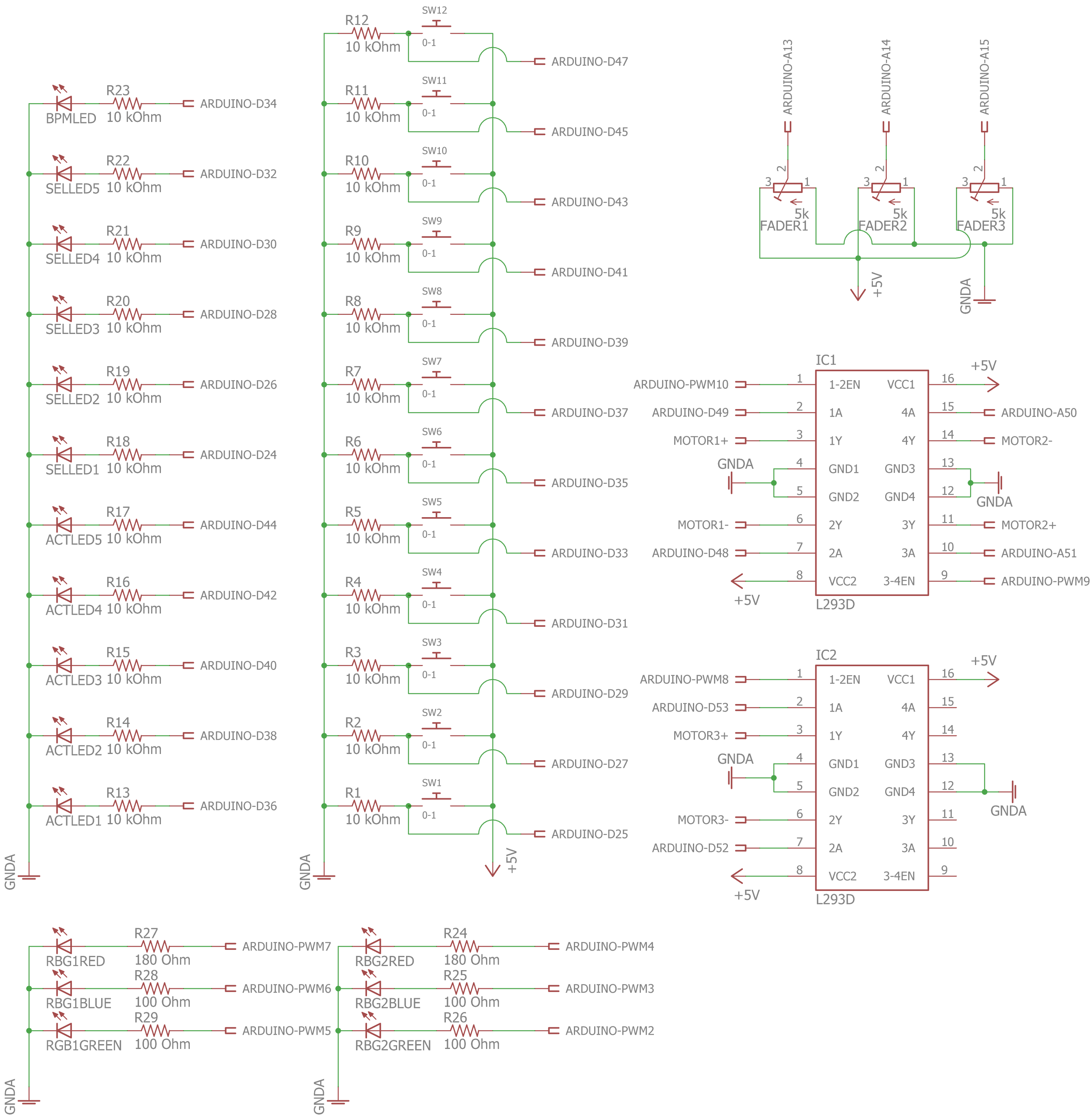

Figure A.6: The schematic of the electrical components as well as their connections in iteration 2.

Appendix B

Semi-structured interview

- Purpose:
  - Does the interface work/make sense
  - Placement of box
  - Sound effects
- Recording
  - Notes
  - Audio
- Questions
  - What is your involvement in the project?
  - Strengths and weaknesses?
  - How would the project change how you play?
  - Sound effects and parameters?
  - What hurdles remain for you to use the project effectively?
  - IMU business
  - What kind of genres would you play with a traditional double bass?
  - How can augmentation expand this?
- Summarized box and effect parameter feedback
  - Dotted quarter and eighth notes for delay.
  - Depth on beat repeat.
  - Reverb should go to 0

- Beat repeat should have possibility to also go in chain
- Maybe possibility to close off for the non pitch shifted signal
- Tempo detection for delay would be insane but probably too complex for now.
- Dry wet on everything, put it on the same slider
- Use IMU for controlling tempo on beat repeat
- Pitch shifter should have different a slider dedicated to octave (-1, +1 +2) and fifth (-1, +1 +2).
- Perhaps we need an extra slider for next iteration that has four sliders.

• Notes:
  - Involvement:
  - Musician, test person, feedback giver, co-developer

• Strength and weaknesses:
  - Anything is possible, we can do anything (within our programming limits)
  - Weaknesses: hard to make it robust, logistic problems. An already made product is easier to just plug in. Other than that no weakness.

• How will it change how you play:
  - Will make him play more double bass, he likes playing but bored of playing the classic double bass.
  - Has to play the same thing, do it correct.
  - Likes to challenge his musicality and the instrument, push the limits of what the double bass can do.
  - You can only do so much with a double bass, it's not very diverse. It's an entirely new instrument, can't call it a double bass anymore.

• Sound effect and parameters:
  - Notes above.
  - Things have improved.
  - Haven't been able to play it yet.
  - He is impressed.

• Hurdles remain to use effectively:
  - Parameters to make it easy, make it special with the effects.

- Has to practice, find out what sounds nice, how it sounds, what to play, how to play, and then practice.
- A new instrument, gotta start from the bottom. Logistic problems, how to put it on the bass.
- If it gets to the point of being finished he wouldn't mind using velcro or something to "permanently" put it on the bass.

- IMU business:
  - Turning on its own axis rather than bending forward.
- Genres on traditional double bass:
  - Jazz and singer song writer stuff.
  - Acoustic pop singer/songwriter things.
  - Mostly jazz though.
- How can augmentation expand genres:
  - Can't use it for pop, he has a role in that band.
  - For jazz, improvisation could be really cool.
  - Bass solo's are muddled sometimes.
  - "Putte sig frem i lydbilledet"
  - Make the sound more produced instead of just acoustic.
  - Make entirely new music instead of bringing into existing genres/music.
- Closing thoughts:
  - Worried about how to get the flow when using it.
  - Thinks this might be a solo thing, standing alone on the stage and do a performance.
  - Might need a loop pedal for this (not in our project scope).

Appendix C

Usability test one

Appendix containing set-up, planning, tasks and results from the first usability test.

C.1 First Usability Test

C.1.1 Planning and set-up

- Introduction
  - Show user how it works
  - Demonstration of effects
  - GUI introduction
  - Familiarisation
  - N minutes of learning
  - Tasks
    - See section C.2
- Free play
  - N minutes of free play
- Scope
  - FX control box
- Purpose
  - Are users able to understand and use the box to activate/deactivate and manipulate effects?
- Schedule and location

  - Week 49, 2016 in SoundLab

- Sessions
  - 1 day in the SoundLab
- Equipment
  - Table, chair, speakers, Max/MSP (on laptop) and FX control box
- Participants
  - Up to 9 SMC students. Recruited through interest of science and research.
- Scenarios
  - A musician wants to turn on/off effects on the fly as well as change their settings.
- Metrics
  - Questionnaires
  - Demographic
  - Understanding after each "item" (Intro, fam. session, tasks, free play.)
  - Overall experience (satisfaction etc.)
- Quantitative metrics
  - Completion time for tasks
  - Time spent in "free play"
- Roles
  - Facilitator / Experiment conductor
  - Note taker

C.2 Tasks

- Task 1:
  - Turn on the delay effect
- Task 2:
  - Increase the feedback of the delay to 0.6

  - Set the rate to 1/8

- Task 3:

  - Turn on the pitch-shifter
  - Increase the level of OctaveUp to 0.6
  - Increase the level of FifthUp to 0.4

- Task 4:

  - Change the rate of the delay to 1/3
  - Change the feedback to 0.8.4

- Task 5:

  - Turn on the reverb
  - Set feedback to 0.58
  - Set damping to 0.3
  - Set Cut-off Frequency to 1500 Hz

- Task 6:

  - Turn on beat repeater
  - Set preset to 2
  - Turn off pitch-shifter

- Task 7:

  - Turn off reverb
  - Turn off the delay.
  - Turn on convolution
  - Set water to 0.4

- Task 8:

  - Select the pitch-shifter (but don't turn it on)
  - Set the level of OctaveUp to 1.
  - Set the level of FifthUp to 0.
  - Set the level of FourthUp to 0.8

- Task 9:

- Turn off beat repeater
- Turn off convolution
- Turn on pitch-shifter

- Task 10:
  - Turn off pitch-shifter

C.3 Questionnaire

- How well did you understand how the control box worked after the introduction / familiarisation phase?
- How well did you understand how the control box worked after completing the tasks?9
- How well did you understand how the control box worked after the free play session?
- How easy was it to activate/deactivate effects?
- How easy was it to change parameters of effects?
- How intuitive was it to change between which effect was being edited?
- How easy was it to understand the different parameters of effects?
- How would you rate your overall experience using the controller?
- How much did you like the Reverb effect?
- Do you have any comments or recommended changes for the Reverb effect? (different parameters, more options etc.)
- How much did you like the Delay effect?
- Do you have any comments or recommended changes for the Delay effect? (different parameters, more options etc.)
- How much did you like the Convolution effect?
- Do you have any comments or recommended changes for the Convolution effect? (different parameters, different impulse responses (water, metal and bees), more options etc.)
- How much did you like the Beat Repeat effect?

- Do you have any comments or recommended changes for the Beat Repeat effect? (different parameters, more options, more presets etc.)
- How much did you like the Pitch Shift effect?
- Do you have any comments or recommended changes for the Pitch Shift effect? (different parameters, different intervals, more simultaneous shifts , more options etc.)
- Over all how satisfied were you with the selection of effects?
- Were there any effects you were missing in the selection?
- Do you have any comments in general about the experience?
- How old are you?
- What is your highest level of education?
- What is your current occupation?

Appendix D

Semi-structured Interview #2

D.1 Questions

The following are the questions we asked during the interview.

- Practice with the box, how was it?
- Performance, how was it?
- Role of the double bass, was it extended?
- Are the audio effects adequate for this purpose?
- Are more than audio effects necessary?
- What hurdles, if any, remain for you to use the project effectively?
- Have you found a genre the project works better for?
- Any other comments?

D.1.1 Practice with the box, how was it?

Cool in many ways, but also a bit troublesome.

Gives a lot of options. Many people think that it's very cool. Setting up is a real hassle.

It's awesome to play with but it's not really a tool for general play.

D.1.2 Performance, how was it?

Only the played at the library (Bag Masken) so far. It was really great using. Once again setup was a hassle but it was great to be able to use it in a performance. Making crazy sounds. Very fun to play along with an electronic musician and sometimes be confused about who was making what sounds. Great to be able to change parameters, having full control! Used only convolution, reverb and pitch-shift which were excellent for the poetry reading. Other projects would definitely be useful to use the other effects too.

D.1.3 Role of the double bass, was it extended?

Last time. Roles: Established that the DB is a rhythm group, so very much the background. Especially, in jazz and symphony orchestra. I really feel that my role as a bass player has been extended - I can now play solo parts and really step out of the background Usually one function, but this box really changes stuff!

D.1.4 Are the audio effects adequate for this purpose?

Yes, but in the beginning I was thinking it was a one-man band thing. But I've mostly be using it as a group thing. Reverb, convolution makes it able to make the craziest soundscapes. It has a great range - everything from nice effects all the way to a powerhouse making a lot of nice.

I don't think the number of effects should be higher. Limitation sparks creations, and you really get to know your tools with a limited amount of effects. If it had more effects it would be unmanageable. It feels like the box has "personality".

If we were to add more effects, I would do it in packs.

- presets of effects
- Different presets for different genres: death metal, jazz, whatever
- Different effects complement each other

A different idea would be to be able to choose effects on your own.

- Drag and drop up to five different effects into the chain.
- Would increase flexibility
- "Swiss Army Box"

D.1.5 Are more than audio effects necessary?

The controls are quite adequate. IMU is actually not that necessary. It's exactly as complex as it needs to be. However, it would be cool if it could be expanded with IMU and stuff like that. Perhaps the rhythmic processing would be cool.Dangerous to add more stuff at this point. Better to refine the product. In a one-man project all expansions would be cool, but I just use it as it is.

D.1.6 What hurdles, if any, remain for you to use the project effectively?

I've already told you but setting up is definitely a hurdle. The number one hurdle, actually. Requires a LOT of gear: stands, cables, soundcard, laptop etc. etc. If not, I'd be able to use it all the time. Takes to long to get started, so I don't practice outside of rehearsals. But when it's up and running - it's super cool and very fun to play. The mounting system is actually not the worst part - can be used as a tabletop controller.

D.1.7 Have you found a genre the project works better for?

Improvisational jazz is what it has been used for with poetry. But the theatre thing is actually a pop-performance.

Not limited to genres - used it a lot for soundscaping. Could really be used in any genre. Electronic universe is an obvious genre because of the sounds. Makes the bass sound quite "Produced" and electronic.. especially considering that it's true acoustic instrument. Would probably not be used in classic music, original jazz. A new element = new genres.

D.1.8 Any other comments?

I'm quite familiar with the controls now. Yeah, I've learned what the different controls do. In a song, when my cue is up, I know exactly what to push. Very intuitive. Especially considering how effects work. Think it's cool I hope you guys will actually finish it. You guys should fix the rest of the issue. Turn it over to an embedded platform. Please do this!

Appendix E

Usability Test 2

E.1 Introduction of Project and Test

The participants were each read the following: "We are in the process of augmenting the double bass by extending its capabilities through electronics and effects processing. All the parameters of the effects can be controlled real-time and for that we have developed a control unit. An example of a parameter is the feedback of a reverb. Reverb is an effect use to simulate large rooms. As you hear, the more we turn up the feedback, the larger the “room” seems."

The test consists of following:

- Introduction to controls
  - Effect activation, selection, parameter control, switching effects, BPM light and BPM change
- Familiarisation
  - Take a minute to familiarise yourself with the controls
- Tasks (performed using the backing track)
  - We’ll give you 11 tasks that you must perform. We will note the completion time
- Questionnaire
  - After the tasks we will ask you to fill out a questionnaire.
- (Optional) Free play with effects either playing the cello or using the backing track.

- We might film you and ask for your thoughts/feedback afterwards

E.2 Tasks

- Task 1:
  - Turn on Effect 2 (delay)
- Task 2:
  - Increase delay Parameter 1 to half
  - Set delay Parameter 2 one step up
  - Set delay dry/wet to 50%
- Task 3:
  - Turn on the Effect 5 (pitch shift)
  - Increase pitch shift Parameter 1 two steps up
  - Increase pitch shift Parameter 2 two steps up
  - Set pitch shift dry/wet to 50%
- Task 4:
  - Change delay Parameter 2 one step up
  - Change delay Parameter 1 to 75%
- Task 5:
  - Turn on Effect 1 (reverb)
  - Set reverb Parameter 1 to 80%
  - Set reverb Dry/Wet to 50%
- Task 6:
  - Turn on Effect 4 (Beat Repeat)
  - Set parameter 1 one steps up preset to 2
  - Set beat repeat dry/wet to 100%
- Task 7:
  - Turn on BPM LED
  - Change BPM to a slower tempo

- Task 8:
  - Turn off effect 1 (reverb)
  - Turn off effect 2 (delay).
  - Turn off effect 4 (Beat Repeater)
  - Turn off effect 5 (pitch shift)
- Task 9:
  - Turn on effect 3 (Convolution)
  - Set Convolution Parameter 1 two steps up
  - Parameter 2 one step up
  - Set Dry/wet to 100%
- Task 10:
  - Turn off effect 3 (convolution)
  - Increase effect 5 (PITCH SHIFT) parameter 1 one more step up without turning it on
  - Increase effect 5 (PITCH SHIFT) parameter 2 one more step up without turning it on
  - Set effect 5 (PITCH SHIFT) dry/wet to 100% without turning it on
- Task 11:
  - Turn on effect 5 (pitch-shift)

Appendix F

Rev3 Object

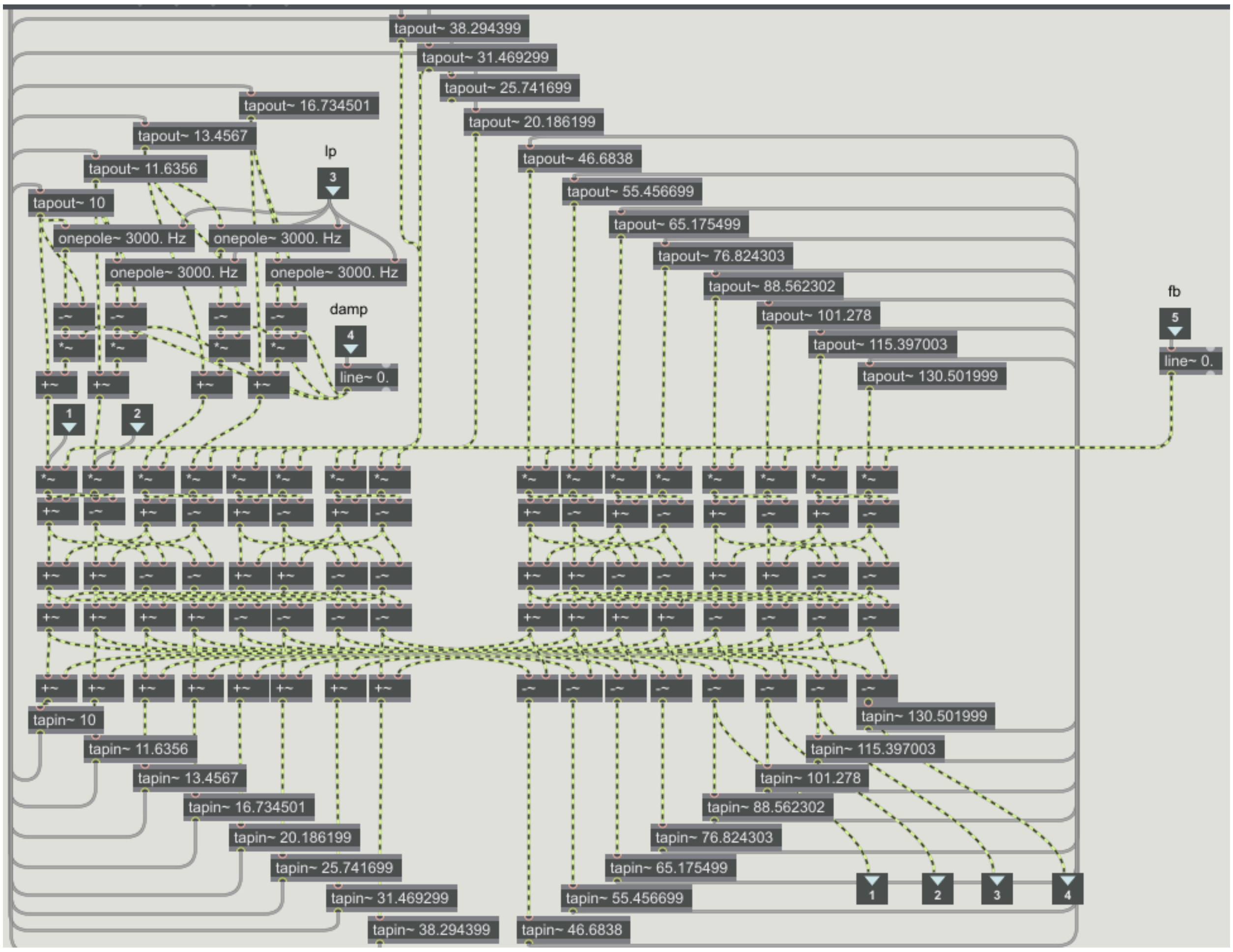

Figure F.1: Rev3 object

Appendix G

NIME Paper

Expanding the Possibilities and Roles for the Double Bass through Augmentation

ABSTRACT

The double bass is traditionally found in a few select genres, playing the same roles it has always played. One way to bring the double bass out of this invisible boundary is through effects and augmentation. This paper aims to present and evaluate a prototype effects box that can control effects specifically tailored to and co-designed by a double bass player. Two evaluation methods are presented that aim to combine the quantitative qualities of a questionnaire for usability design and the qualitative qualities of including the performer in the design process from the very beginning.

Author Keywords

NIME, augmentation, double bass, audio effects, digital signal processing

ACM Classification

H.5.5 [Information Interfaces and Presentation] Sound and Music Computing, H.5.2 [Information Interfaces and Presentation] User Interfaces—Auditory (non-speech) feedback, Prototyping, Evaluation/methodology, Haptic I/O

1. INTRODUCTION

In a musical performance you often find that instrumentalists such as guitar players have a plethora of effect pedals. Each pedal is placed in a chain in an order that makes sense effect wise. Each individual pedal has a set of knobs to control parameters of the pedal; the parameters of a reverb could be feedback, cut-off frequency for reverberations and damping. The effect pedal itself can then be turned on and off by stepping on a switch with your foot. However, changing the knobs with your foot is highly impractical and makes parameter control all but impossible without having to bend down to the actual pedals. Our project is done in collaboration with a double bass player who wishes to use audio effects on his double bass but retain the possibility of changing parameters on the fly. However, as he is usually playing the bass while standing it makes it even more of a challenge to manually change parameters real-time on regular guitar pedals. And so, we set out to explore the possibilities for alternate controls.

NIME'17, May 15-19, 2017, Aalborg University Copenhagen, Denmark.
.

A short video introduction and demonstration can be found by following the link in section 8.

2. BACKGROUND

In a previous unpublished project, the authors researched convolution and ways to simplify and enable anyone to use this technology in new novel ways. The project also focused on using different signals than room impulse responses for convolution: for instance a plucked cello string or water be poured into a glass. During the testing phase a double bass player tested it and immediately his thoughts went to how he could use convolution on his double bass to create an entirely different experience than the traditional double bass play. He then asked us to go a step further and make a dedicated augmented double bass. This leads us to this paper in which we attempt just that. However, one must look at previous augmented instruments to get an idea of what the state of the art is. There are very few (if any) instances of a double bass being augmented, but there are still many things one can learn from any instrument being augmented, as such, this section will specifically look at what has been done before in terms of augmenting an existing instrument. What follows is a range of different augmented instruments.

2.1 The Augmentalist

The Augmentalist is an easy to use, user-centred system which allows any musician to augment their instruments with ease. The sensors are Phidget sensors which include buttons, FSR's, faders and accelerometers that are easily attached to the instrument of choice [7]. The mapping and interpretation of the sensor data is handled in Max/MSP [3] where it is converted into MIDI signals. The MIDI signals can then be used to control whatever audio software the user desires. For this project one of the most interesting aspects about the Augmentalist project is the close collaboration with musicians and the idea of "musicians as developers"; especially, since they are the experts on their instruments and thus know which augmentations are feasible or not.

2.2 Hybrid Violin

An example of a more instrumental specific augmentation is the hybrid violin [11]. This paper concerns their augmented electric violin [11] which they have augmented with sensors and an iPod touch that uses these sensors to control a pd patch that runs on the iPod via MobMuPlat. The important design elements were as follows: High mobility in a battery powered approach, all processing done via the violin platform, interfacing done via the iPod Touch and high accessibility through a low cost electric violin. The hybrid violin was tested on a string quartet in which they had three violinist and a cellist. The approach was a semi structured interview, i.e., highly qualitative information. The audio quality and loudness was a problem. For gestures/mapping there was interest among the test subjects, but more as a novelty as it lacked exploration, more complex mappings and additional sensings were necessary. It was also important to design the instrument for a specific target group: is it a solo instrument or a group effort? Overholt was very focused and precise in both description of the design and evaluation, however, at the time the Hybrid Violin was still very early in its development.

2.3 Hybrid Piano

Another instrument specific augmentation example is the hybrid piano [4]. Dahlstedt's augmented hybrid piano [4] consists of a piano and a sound processing unit with speaker and microphones placed such that the acoustic and processed sound blend into one, he calls this Foldings. The processing techniques include virtual resonance strings, dynamic buffer shuffling and acoustic and virtual feedback. The instrument builds upon the foundation that there should be correlation between the physical effort exerted playing the instrument and the sound produced. It should be as free and direct as an acoustic instrument with no extra faders or knobs. Evaluation is done via his own qualitative experiences and one other pianist. He argues that this is the necessary approach as one cannot evaluate a complex instrument unless one builds up experience with the instrument over several years. This approach is about trying to understand the possibilities of the instrument rather than saying "this instrument is better than that one" as he puts it. While the paper is interesting it is very far from a double bass, both in terms of sound but also simply the way the instrument is played.

2.4 SABRe

The bass clarinet has more in common with the double bass in terms of the frequency content but also the fact that both hands are occupied which is an important aspect when designing an augmented double bass. Therefore, the SABRe was looked at. The authors explore the possibilities of augmenting a bass clarinet to extend the possibilities within performance and composition. The project SABRe [12] (Sensor Augmented Bass clarinet Research) includes a plethora of sensors which are processed using dedicated PSU's which are then sent to a computer using OSC. Their goal was to extend the player's possibilities of manipulating effects and signal processing through gestures and sensors without compromising the skills already acquired by the player. The last part is an important aspect for this project as the hands of a double bass player are very much occupied when playing; it is important to consider and ensure that the player has easy access to and controls of the desired effects.

2.5 Summary

Many of the design aspects that can be found in the previous examples are useful. An important aspect that can be found in both the hybrid piano [4] and the Augmentalist [7] is including the musician in the design process from the start: the tool-maker is also the tool-user as Dahlstedt [4] puts it. Another aspect is the evaluation method which will be explored more in depth in Section 4. Furthermore mapping and control are important aspects but as none of the above projects work with double basses we will need to work closely with our collaborator to find a feasible set of controls.

3. DESIGN

As described in the introduction our challenge was to find a way to give a double bass player access to effects and furthermore the ability to change effect parameters on the fly. Our mission is then to augment a double bass so that its capabilities are extended through the use of sensors, sound production and new modes of interaction. The sound is captured using an existing pick-up on the bass while an Arduino captures the sensor data and sends it to a computer for audio processing. To come up with ideas and explore aspects we had not considered before, we did a Verplank [15] sketch which can be seen in fig. 1.

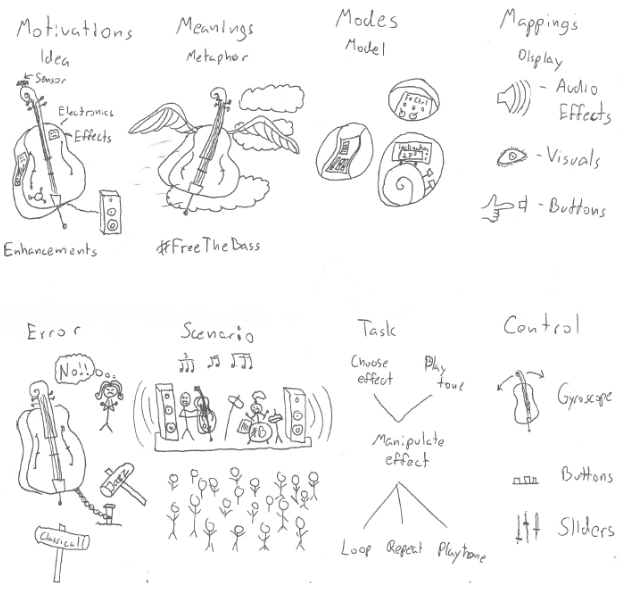

Figure 1: Verplank sketch showing the initial ideas for augmenting the double bass.

Going through the Verplank sketch we find that, in our motivations, we set out to extend the double bass and its inherent sonic features. This done through digital sound processing on a computer which is controlled through sensors. The reason for wanting to do this is that the double bass is mostly seen in genres such as classical, jazz and folk music. We wish to free the double bass and enable it to be a part of performances where you would not usually find it, enhance the role in these genres or enable musicians to create entirely new genres that were not possible before. For instance, our collaborator found that solo pieces by double basses can be very diffused in their sound; being able to put yourself forward in the soundscape was interesting to our collaborator (see 4.1). The task is to give the player easy access and the ability to change audio effects. Controls of effects are made available through sensors such as faders, push-buttons and even a gyroscope.

Before settling down on the first iteration (see subsection 3.1) other ideas were considered. The first thing that springs to mind is the use of a tablet. However, it was quickly discarded due to the necessary placement of the controls. In short, it is hard to control a touchscreen when you can not see it as it has no tactile feedback. Another idea that was considered was the use of contact microphones to enhance the percussive elements that already exist in the double bass. One could, for example, drum a beat on the double bass and loop it.

3.1 Design Choices

The initial idea for a control unit consisted of buttons and faders through which effect activation and manipulation is possible. The initial idea is sketched in fig. 2 in which we

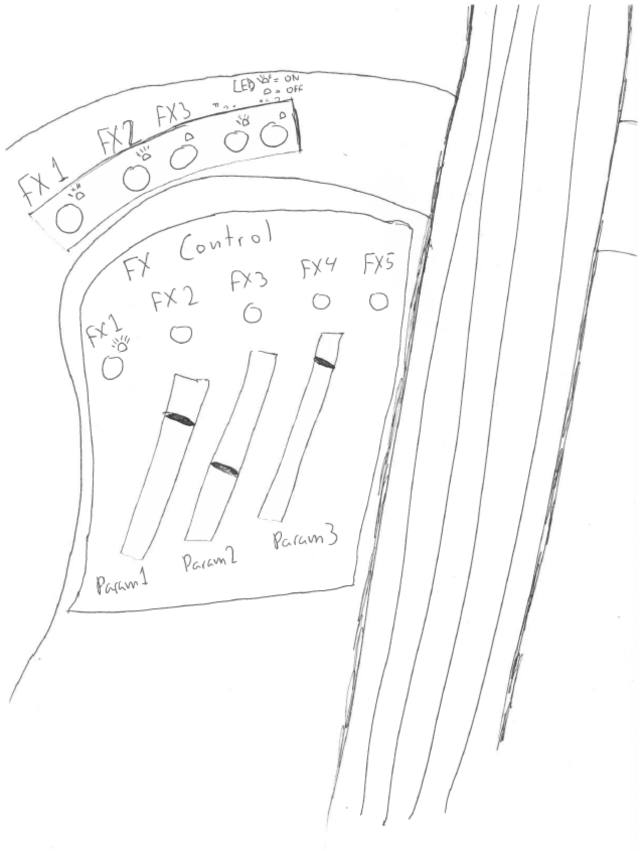

Figure 2: Sketch of the initial idea - minus the placement.

have one row of buttons to activate/deactivate each individual effect. The other row of buttons would then select which effect to manipulate with the faders. This solution will allow the user to change parameters on any effect without having to turn it on or off.

Double bass players are limited in terms of time to use their hands on anything but playing the double bass. While they can turn on and off effects and change parameters between movements in music or really slow passages, a completely different way of controlling effects during play was needed. To this effect, a gyroscope was considered which allows for control of effects via leaning the instrument in one direction or the other. A notable challenge is the weight and size of the instrument. One is only able to lean a double bass forward so much before the angle makes it unplayable or the sole weight of the double bass is too much to endure. This means lowering resolution of the effect's min/max values. At the same time, the augmentation should impose as little as possible on the player's skill and play style. As Nicolls [8] notes in her study, subconscious movements naturally occurring in a performance could, if used to generate data for audio feedback, disrupt the artistic performance and change focus to controlling the sensors. Therefore, either a deadzone in which the gyroscope sensor data would not be used or a switch to turn on IMU sensors was necessary.

We quickly reached a consensus with our collaborator on the placement of the control unit.

This space is the most easily reached place on the double bass when not strumming (see Fig. 3). The interface itself should be easily used with just tactile feedback or at a glance down. Hence, the initial design features two rows of buttons and three faders. Each row should be on a different side of the bass and each button should have a diode lighting up when active. Three faders control the parameters of the most recently pressed effect. Precise and haptic feedback was deemed necessary if the user is to configure parameters without looking. With this in mind, motorized faders were used for this purpose.

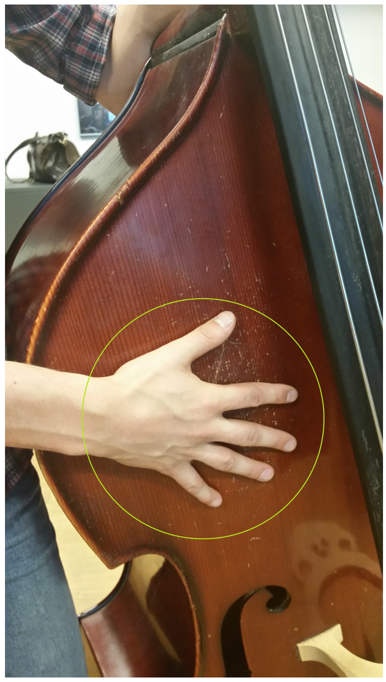

Figure 3: Control space for the UI.

3.2 Foot Pedals versus Hand Controls

As can be conferred in the introduction, it is a major change to move effect control from the established foot pedal to a hand controlled effects box. The obvious argument against hand controls is that you need free hands to play your instrument. However, if a high number of effects are wanted, having three pedals for every effect (assuming three parameters per effect) would fill up a lot of space. And because of the double bass, any movement is heavily impeded, especially while playing. This means unlike guitar players who can move around on stage, double bass players cannot as easily operate a large array of effect pedals. This is then the advantage of hand controls as you are able to finely control three parameters for five different effects on a relatively small box.

3.3 Sound effects

The sound effects are intrinsically connected to the success of the box. Therefore the collaborator was involved in choosing which effects and parameters he felt was the most important. It was an iterative process in which we presented the effects for him to review. The current list of sound effects and parameters is as can be seen in Table 1. See [10] for more details on some of the effects. A sixth effect is a

Table 1: Sound effects and parameters

Effect	Parameter 1	Parameter 2	Parameter 3
Pitch shift	Octave (-1 +1 +2)	Fifth (-1 +1 +2)	Dry/wet
Delay	Feedback	Rate	Dry/wet
Convolution	Impulse sound (1-5)	Sound bank	Dry/wet
Reverb	Feedback	Dampening	Dry/wet
Beat repeater	Preset (1-5)	Filtering type	Dry/wet

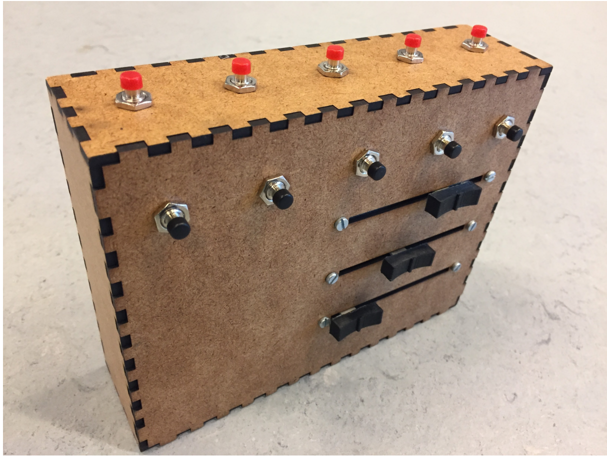

Figure 4: The control box for effects.

possible future extension for the prototype which could be controlled by the IMU on a mobile phone. This effect is flexible and could control many possible effects such as controlling the beats per minute (BPM) on the beat repeater or the delay or an entirely new effect such as distortion.

3.4 Prototype

The original idea is a sleek and flat interface you could easily mount on the bass. But before the project can reach that stage, prototyping must take place. The current prototype is a wooden box big enough to house all electronics we are working with. Likewise, all the controls are also fit into this one box for the sake of simplicity and to make it easier to build. This box is then supposed to be attached to a double bass.

3.4.1 Hardware

The box contains 10 simple push buttons and 3 motorised faders. The box can be seen in Fig. 4. The red buttons on top are used to activate and deactivate effects. The black buttons on the front are used to select which effect is being manipulated. The three faders are used for changing the chosen effect's parameters. To ensure that the faders are usable for all effects, motorised faders were chosen in order to restore previous settings of effects when switching between them. The box also contains an Arduino Mega, which is used for reading the values/input from both buttons and faders. The prototype also utilizes a phone for its IMU capabilities. An application called TouchOSC enables OSC messages to be transmitted directly to Max/MSP [3].

3.4.2 Electronics

Even though all sensor data is being read directly by the Arduino we decided to gather all electronic connections into a board within the box and from there connect them to the Arduino as can be seen in fig. 5 - all sensors are wired to the board where they are given VCC and GND as required. The board also sports a pair of L293D's (H-bridge) that are used to control the fader motors using PWM. The choice to have the microprocessor easily removable was to make any changes in the prototype that much easier to make. With our current design, we are also able to easily replace the Arduino and even replace it with a different processing unit. The setup is sketched in Fig. 6.

3.4.3 The Box

The box is cut from a 3mm MDF board in the dimensions 180mm x 140mm x 50mm (length x width x height) using an

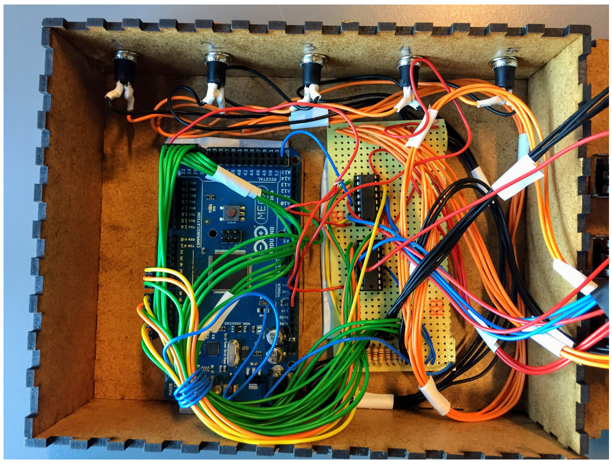

Figure 5: Inside the control box

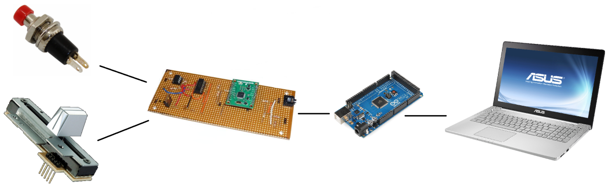

Figure 6: Overview of the electrical components and their connections.

Epilog 30 Fusion Laser Cutter & Engraver. The design was created using MakerCase [5] where the desired dimensions are input whereafter a SVG-file is created. We chose to do this because the box was easy to assemble due to the finger edge joints. The dimensions were based on the amount of space needed by the electronics board, the Arduino and the wiring.

3.4.4 Software

All sensor data captured by the Arduino is sent to Max/MSP [3] using Firmata [13] code which essentially overwrites the OS of the Arduino and facilitates the Maxuino [6] software which we use within Max/MSP. Max/MSP is a visual programming environment in which all sound processing is done. The GUI available to the users can be seen in Fig. 7. The GUI shows the user which effects are activated, which parameters are set for each effect and also which effect is selected (i.e. which effect is currently being manipulated). It is also possible for the user to either set a BPM or tap in the desired tempo. It is important to note that the GUI is not going to be available to the performer when he is on stage. Rather, the GUI was to make it possible to do the usability test and it has the prospect of being used to fine tune effects for the user when not performing.

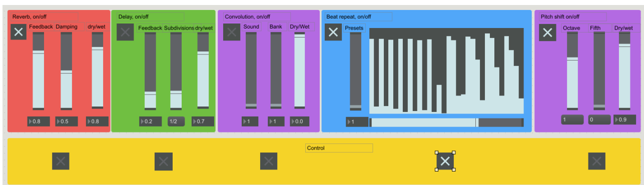

Figure 7: The Max/MSP GUI

4. EVALUATION

The ever-present dilemma in evaluating NIMEs and projects like ours is sufficient quantitative data versus appropriate time given to the performer to learn the instrument. As Dahlstedt [4] mentions; a musician builds up accumulated expert knowledge over many years of playing an instrument. This kind of evaluation is important as "quantitative lab tests can hardly tell us if an instrument works out there, with expert co-players, in front of an audience". This paper utilizes a collaborator in the form of a double bass player to account for this. This is done through continued dialogue during the entire project, and through longer evaluation periods using the "long game" [9] method which resembles the time it takes to learn a new piece of music, with a semi-structured interview at the end. A second experiment was done with the box itself running a double bass backing track. It uses context based tasks as described by Modhrain [9]. This test was done purely to test the usability of the box and the effects.

4.1 Expert test

The expert test was conducted in a practice room at the royal academy of music in Aalborg. One person conducted the experiment with two participants: the collaborator and his double bass teacher. The participants were shown how the box functions after which they recorded a piece on a double bass and used it for the box. Each effect were used and commented upon, after which a semi-structured interview was conducted with the collaborator only. The different topics included strengths and weaknesses, how would the project change how you play, comments on sound effects and parameters, what hurdles remain for you to use the project effectively, comments on IMU, what genres does he currently play with a double bass, how can augmentation expand this.

4.1.1 Results

The collaborator mentioned when discussing the different hurdles to overcome before being able to use the product effectively, that he would definitely need to practice with the instrument (because the augmentation makes it a different instrument altogether) and compose a music piece specifically for the augmented double-bass. Therefore, as there were no outstanding problems with the design of the box, he could not comment in depth on it. However, many of the effects needed either new parameters or changes to their existing parameters such as dotted eights for the delay rate. Another problem identified was the IMU motion . Rather than bending forward or sideways it was deemed a much easier motion to simply turn.

4.1.2 Second expert test

In a second smaller expert test, the device was mounted on the double bass. A pickup microphone through a soundcard connected to the computer provided sound. The collaborator tested the various effects again (done after changes to the effects were made from testing). The main points we took away from this test was the necessity of a more robust mounting system than simple elastic ropes for bicycles and adhesive putty (see Fig. 8). The surface of the double bass is not entirely flat, so a redesign of the box is also necessary. A few solutions were considered such as a rig similar to that of a violin's neck brace. Another solution included using foam on the bottom of the box to fit the curved form of the instrument to a higher degree.

4.2 Usability test

Figure 8: The mounted device.

A usability test was conducted to test the control box on its own where 7 students of the Sound and Music Computing master's programme at Aalborg University. participated in the test. The test consisted of 3 phases followed by a survey for the participants to provide feedback. The test's tasks were is based on guidelines by the `usability.gov` website [14] while the questionnaire was based on a paper by Bin, S Astrid et al.[2]

4.2.1 Method

In the first phase, participants were introduced to the controls and a GUI (the same GUI that is seen in Fig. 7). After the introduction, they were then given a few moments to familiarise themselves with the controls and the GUI. In the second phase the participants were asked to complete 10 tasks which included both activation/deactivation of effects, as well as changing parameters of the effects using the box. Throughout the tasks, a background track of a double bass was playing in order to provide audio feedback for the participants. In the final phase, participants were allowed to freely experiment with the controls and effects on the same background track from the tasks. After the test, the participants were asked to answer a survey where they had to rate a number of attributes on a scale of 1 to 7 along with optional comments on controls and effects. The rated attributes included: understanding of the box after each phase, understanding of effect parameters, satisfaction with the effects and overall satisfaction with selection of effects.

4.2.2 Results

In analysis of the results we can see that there is a slight increase in understanding from phase 1 (introduction and quick familiarisation) to phase 2 (tasks). In phase 3 there is a slight and puzzling decrease (see Table 2). There could be a number of reasons to this. One being that the user had no clear goal laid out in front of them on a piece of paper. This might create situations where they do not know what exactly to do.

Table 2: Participant's understanding of controls

Understanding	Phase 1	Phase 2	Phase 3
	6.71	6.86	6.43

In Fig. (9) We see overall high scores. The figure shows how easy it was to activate and deactivate effects, change

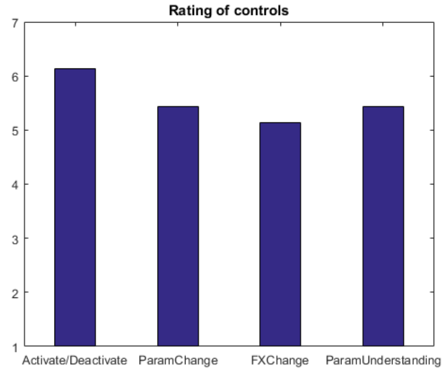

Figure 9: Ratings for Control

parameters, change which effect you are controlling and lastly how well the participants understood the parameters for the effects. The lowest score (FXChange) is understandable as it can be less intuitive to change which effect you are controlling but not actually turning anything on or off.

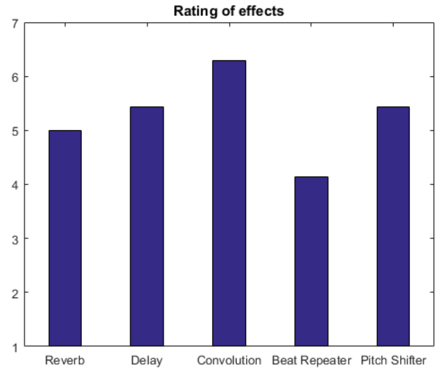

Figure 10: Ratings for sound effects

In Fig. (10) we see the scores for all effects. The more novel convolution effect received the highest scores while reverb and beat repeat lack behind. Based on the comments, these effects were changed. Reverb was altered so that smaller changes in the feedback parameter would have greater effect. The cut-off frequency parameter was also changed to dry/wet mix instead. The beat repeat effect was changed to affect 100% of the sound mix rather than the previous 50%. Upon completion of the next iteration, a second usability test is planned so that we can compare the scores of the first iteration versus the second.

5. FUTURE WORK

There are many possible future iterations do be done on this project. Fine tuning the effects and parameters alone is worthy of multiple iterations. New and more novel effects can also be considered to further expand the role of the double bass such as rhythmic processing. The box itself can move towards the initial design of a sleek interface that is less cumbersome than a box. The method of communication between box and processing unit (a computer in the prototype) will also undergo some changes. A Morpheus board [1] has been acquired and it will be investigated whether or not it can reliably do the same things the laptop does for the project. Cutting away the need for a laptop will more effectively cement the the project as a more finished product.

6. CONCLUSIONS

Based on the response of two double-bass players we can conclude that there is definitely a potential to have real-time control of effects and their parameters. The buttons were fast and responsive and the motorised faders work well both for the user to control effect settings and to get feedback on the current settings of an effect. Upon mounting the device, we found that it is entirely possible to control it, and much faster than we initially thought, however, the current design is not well suited for the curved surface of the instrument. A new mounting system is necessary. The availability of effects also changes the way the instrument should be considered and played as it introduces new possibilities as well as challenges.

7. REFERENCES

[1] BENTI. BENTI - DIGITAL AUDIO R&D http://www.bentidigital.com/morpheus-board.html, 2016.

[2] S. A. Bin, N. Bryan-Kinns, and A. P. Mcpherson. Skip the Pre-Concert Demo: How Technical Familiarity and Musical Style Affect Audience Response. *NIME '16 Proceedings of the 2016 Conference on New Interfaces for Musical Expression*, pages 200–205, 2016.

[3] Cycling'74. Max https://cycling74.com/.

[4] P. Dahlstedt. Mapping Strategies and Sound Engine Design for an Augmented Hybrid Piano. *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 271–276, 2015.

[5] J. Hallander. MakerCase - Easy Laser Cut Case Design, http://www.makercase.com/, 2014.

[6] Maxuino. Maxuino http://www.maxuino.org/, 2010.

[7] D. Newton and M. Marshall. The augmentalist: enabling musicians to develop augmented musical instruments. In *Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction*, pages 249–252. ACM, 2011.

[8] S. Nicolls. Seeking Out the Spaces Between: Using Improvisation in Collaborative Composition with Interactive Technology. *Leonardo Music Journal*, 20:47–55, 2010.

[9] S. O'Modhrain. A Framework for the Evaluation of Digital Musical Instruments. *Computer Music Journal*, 35(1):28–42, 2011.

[10] S. J. Orfanidis. *Introduction to Signal Processing.* Pearson Education, Inc., 2010.

[11] D. Overholt and S. Gelineck. Design & Evaluation of an Accessible Hybrid Violin Platform. *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 122–125, 2014.

[12] S. Schiesser and J. C. Schacher. SABRe : The Augmented Bass Clarinet. *NIME 2012 Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 109–112, 2012.

[13] H.-c. Steiner. Firmata: Towards making microcontrollers act like extensions of the computer. *New Interfaces for Musical Expression*, pages 125–130, 2009.

[14] Usability.gov. Running a Usability Test, 2014.

[15] B. Verplank. Interaction design sketchbook. *Unpublished paper for CCRMA course Music 250a*, pages 1–23, 2003.

8. VIDEO ABSTRACT

https://youtu.be/9Cz5o6rsAHE

Appendix H

Results from Usability Test 1

The following are the results from the first usability test that was conducted during the first iteration.

#	How well did you und	How well did you unc	How well did you unde	How easy was it to activate/
f082a4c0c	6	7	7	6
eb9f030af	6	7	7	7
018000cd	7	7	7	7
779c3d58	7	7	4	3
08423307	7	7	7	6
d35e11c2	7	6	6	7
055dc661	7	7	7	7
Avg	6.714285714	6.857142857	6.428571429	6.142857143

	Understanding (p1)	Understanding (p2)	Understanding (p3)	
	6.71	6.86	6.43	
	Activate/deactivate	Parameter change	Effect change	Parameter understanding
	6.14	5.43	5.14	5.43
	Reverb	Delay	Convolution	Beat repeater
	5.00	5.43	6.29	4.14
	Effect selection	Overall experience		
	5.71	6.14		

How easy was it	How intui	How easy	How woul	How mucl	Do you ha	How mucl	Do you have any comment	How mucl
6	5	5	6	5		5		7
5	4	4	5	3	I found th	5	Add BPM in there.	7
6	6	7	6	6	Size of the	7	Change the type of delay (t	5
2	5	6	6	6		6		7
7	3	4	6	4		5		7
5	6	6	7	7	no	6	no	5
7	7	6	7	4	Wet/dry r	4	a little bit like with the reve	6
5.428571429	5.142857:	5.428571‹	6.142857:	5		5.428571‹		6.285714:

Pitch shifter	
5.43	

Do you ha	How mucl	Do you ha	How mucl	Do you ha	Over all h	Were there any e
	3		6		6	No
More imp	2	Didn't rea	7	LOVED IT.	6	Phasing/flanging
no	4	No	7	No	6	Distortion, filters
	4	can't unde	4		4	no
I loved wa	7	I loved it b	5		6	No
no	6		5		7	
It was ver	3	I felt like t	4	Many way	5	Wah, Flanger, Ste
	4.142857		5.428571		5.714285	

Do you have any comments in general abou	How old a	What is yo	What is yo	Start Date	Submit Da	Network I
No	24	Bachelor's	Student	2016-12-0	2016-12-0	02210b40
My hands were sometimes in the way for th	22	Bachelor's	Student	2016-12-1	2016-12-1	0f437ce8e
One thing that sometimes caught me off gu	23	Bachelor's	Student	2016-12-1	2016-12-1	0f437ce8e
color catagory	29	Master's d	Student	2016-12-1	2016-12-1	0f437ce8e
I had to think very carefully when switching	27	Bachelor's	Student	2016-12-1	2016-12-1	0f437ce8e
It was fun to use	23	Bachelor's	Student	2016-12-1	2016-12-1	0f437ce8e
I know that it is a prototype, but labels wou	26	Bachelor's	Student	2016-12-1	2016-12-1	0f437ce8e
	24.857142					

Appendix I

Acknowledgements

We thank Jesper Rindom Jensen for excellent project supervision throughout the project. We would also like to thank Mads Thorlund Rømer for inspiring cooperation in realising this project and Torben Bjørnskov double bass expertise. We thank Thomas Kristensen for advice on electrical circuitry. We thank Markus Lčhtefeld for advice and guidance on sensors. We thank Poul Lund, Peter Skotte and Jens Munk Clemmensen for advice on mounting the controller to the double bass.

Appendix J

A brief guide to the BAES

This appendix briefly describes the requirements for running the BAES as well as guide to setting up and running the system.

J.1 Software requirements

- Windows 8/10
- Arduino IDE
- Max 7 32-bit
- (Soundcard specific drivers)
- The latest version of the BAES Max project

J.2 Hardware requirements

- The BAES controller and Arduino box
- USB cable type-B

J.3 Setting up

This is a step-by-step guide for running the BAES. It is important to go through the steps in the order.

1. Attach the BAES controller and Arduino box to the double bass
2. Connect the external sound card (if using one)
3. Connect the Arduino box to the laptop with the USB-B cable

4. Open Max 7 32-bit
5. Select File –> Open...
6. Browse the Max project in the prompt and open BAES.maxproj
7. Turn on Max's DSP
8. Activate Direct Input
9. Set input controls to an appropriate level using the gain slider

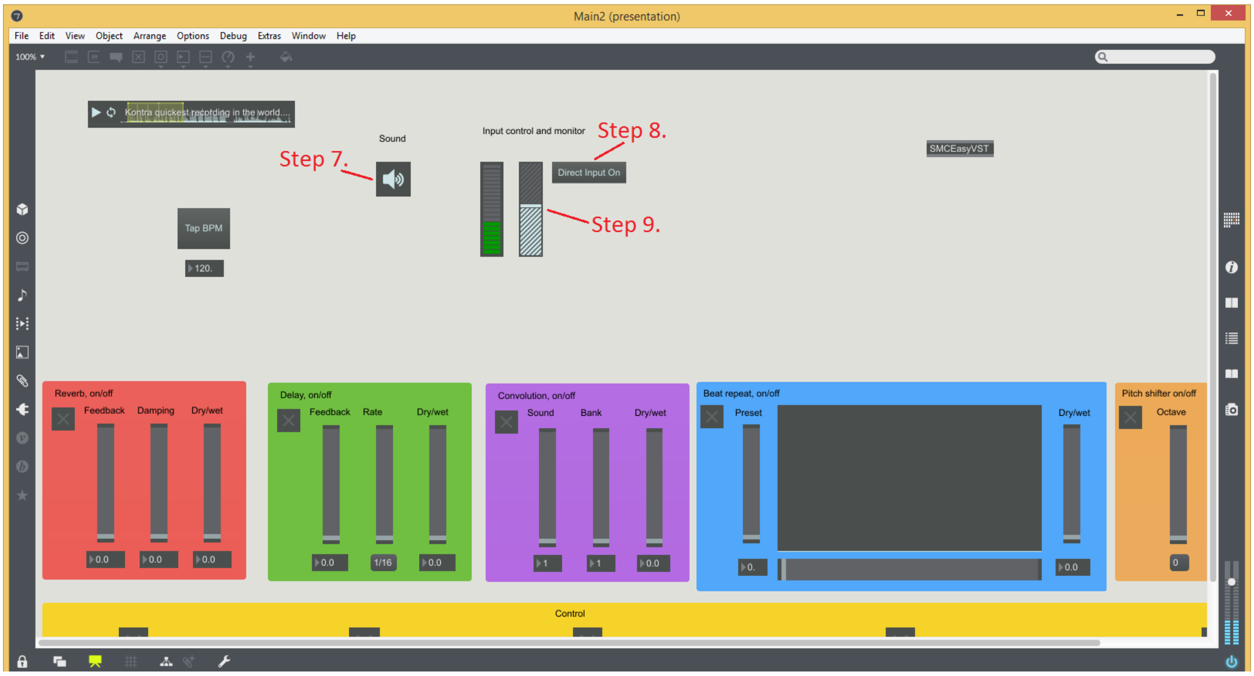

Figure J.1: Diagram overview of data flow

If each step is completed in the correct order, the BAES should be functioning. Make sure to turn on Max's DSP, activate Direct Input and set the input gain to an appropriate level.