AUDITORY VIRTUAL ENVIRONMENT WITH DYNAMIC ROOM CHARACTERISTICS FOR MUSIC PERFORMANCES

By

Daniel Dhaham Choi

A Thesis Submitted to the Graduate

Faculty of Rensselaer Polytechnic Institute

in Partial Fulfillment of the

Requirements for the degree of

MASTER OF SCIENCE

Major Subject: ARCHITECTURAL SCIENCES

Approved:

_____

Dr. Jonas Braasch, Thesis Adviser

_____

Dr. Ning Xiang, Committee Member

_____

Dr. Ted Krueger, Committee Member

Rensselaer Polytechnic Institute
Troy, New York

June 2013
(For Graduation August 2013)

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

A room-adaptive system was designed to simulate an electro-acoustic space that changes room characteristics in real-time according to the content of sound. In this specific case, the focus of the sound components is on the different styles and genres of music. This system is composed of real-time music recognition algorithms that analyze the different elements of music, determine the desired room characteristics, and output the acoustical parameters via multi-channel room simulation mechanisms. The system modifies the acoustic properties of a space and enables it to "improvise" its acoustical parameters based on the sounds of the music performances.

1. INTRODUCTION

1.1 Artificial Acoustics

Many people use the term “artificial acoustics” in reference to electro-acoustics. However, Barry Blesser describes artificial acoustics as any form of acoustical modification with the purpose of achieving a desired sound, without necessarily including the use of electronics (Blesser & Salter, 2007). According to Blesser, artificial acoustics exist even in concert halls without electronics, when any form of acoustical treatments is used. Thus, in regards to the topic of artificial acoustics, this paper is based on the electro-acoustic system.

1.2 The Electro-acoustic System

An *electro-acoustic system* is a system that uses electronics in order to alter sound, whether it is through amplification, distribution, or reverberation. The history of acoustics dates back to the Greek amphitheaters, with electro-acoustics being developed in the late nineteenth century (Blesser & Salter, 2007). At the early stages of electro-acoustic systems, the desired sound emphasized clarity, and reverberation was considered to be an unpleasant noise. However, in the middle of the twentieth century, reverberation started to become a desirable feature in sound and society began perceiving it to be aesthetically pleasing.

1.2.1 Analog Simulation of Reverberation

The first form of simulating reverberation in the electro-acoustic system started in the studio recording industry (Blesser & Salter, 2007). Since at the time recorded and broadcast sound was fairly dry, sound engineers explored methods in which ambience and artificial reverberation could be incorporated into their sound. The reverberation chamber was their first solution. It is a reverberant room consisting of a speaker and a microphone. The speaker would play the dry sound, and the microphone would pick up the sound reverberated in the room (Rettinger, 1957). Although this was a groundbreaking technology, it did not have the sound of concert halls. Most

reverberation chambers were too small and had strong resonances; furthermore the reverberation was not variable.

Another simulation method for reverberation was the spring reverberator (Blesser & Salter, 2007). It consists of a metal spring, and has transducers at both ends. One transducer passes the signal vibrations into the spring; the time taken for the vibrations to travel along the spring causes multiple delays. These delays, collected by the transducer at the pickup end, are mixed back in with the direct signal to create an artificial reverberation effect. The spring reverberator at the time was popular for its compactness and low cost. It was also capable of being used in portable devices such as guitar amplifiers or Hammond organs. People had also used the feedback loops of tape machines with the spring reverberator to generate echo effects.

Then, in 1960, the plate reverberator was developed (Blesser & Salter, 2007). It consists of a large thin steel plate with two transducers, similar to the spring reverberator. Unlike the spring reverberator, the plate vibrates along two axes thus reducing coloration and limitations in reverberation time. It was also considered to be a higher quality concert hall simulator because it adds less noise. Many models have control over reverberation time with an attached damping pad; the distance between the damping pad and the plate is proportional to the reverberation time.

Another popular analog delay system was the bucket-brigade device (BBD) developed in 1969 (Sangster & Teer, 1969). It stores analog information in very small capacitors and releases it with different delay times through different charge transfer arrangements controlled by a clock signal.

1.2.2 Digital Simulation of Reverberation

As the digital era slowly began to dominate technology towards the seventies, reverberation was generated digitally as well. The first form of digital reverberation was the algorithmic reverb. This consisted of programmed algorithms that processed the dry signal through multiple parameters (such as delays and filters) to simulate reverberation. The concept of Feedback Delay Networks (FDN) was used to create a feedback by connecting the output of each delay line to its input (see Figure 1.1). This increased the

density of the reflections and also increased modal density by distributing the energy of the modes throughout the different delay lines (Smith & Lee, 2007). Through these methods and algorithms, the sounds of concert halls, rooms in various sizes, and even the analog reverberation simulators (such as chamber, plate, spring) were simulated.

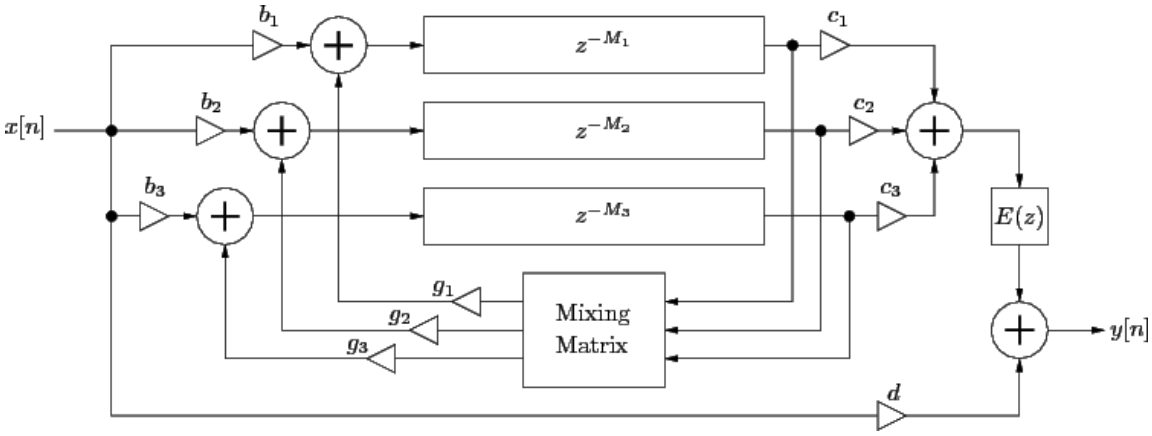

Figure 1.1: An example of a structure of the feedback delay network. Image taken from (Jot, 1992).

There were commercial products that digitally generated reverberation in numerous recording studios; hardware systems such as the Lexicon 480L were and still are popular to this day. With the development of digital audio workstations, reverberation was processed entirely in the digital domain through plugins and software. With the advancement of computational speed and processing power, the convolution reverb was developed. It takes an impulse response, measured or generated, and convolves it with the dry signal, consequently simulating the reverberation of the space.

1.2.3 Acoustic Enhancement

Digital reverberation expanded beyond the studios to concert halls and other live performance venues. An *acoustic enhancement system* is a form of an auditory virtual environment that controls the acoustical characteristics of a space. This includes manipulating the energy and reflections of the reverberant sound, and therefore requires a multi-channel speaker system. Commercially available acoustic enhancement systems

include the Meyer Constellation, the Yamaha AFC3, and the E-coustic LARES. These multi-channel systems consist of multiple microphones and speakers placed throughout the space. The system creates a virtual acoustical space through digital signal processing, capturing the sound in the space with the microphones and simulating the room's acoustical responses through the multi-channel speakers (Griesinger, 1991).

Figure 1.2: System configuration of the Yamaha AFC3. Image taken from (Yamaha Corporation, 2013).

Acoustic enhancement systems such as the E-coustic LARES feature microphone placement reasonably distant from the source (Griesinger, 1991). The LARES is known for adequately capturing the source sound and producing high reverberation levels without coloration due to acoustic feedback. However, the system designed for this thesis does not require the microphone to be distant from the source, as it is does not receive its sound signal from microphones fixed in the space.

1.3 System Adapting to Music Performance

As music performance spaces become less concentrated to one style of music and more diverse in their uses, the versatility of their acoustical characteristics is crucial to a modern performance space. Many halls today feature variable acoustics to be able to accommodate different styles of music. A notable example at the beginnings of variable acoustics is the Espace de Projection at the Institut de Recherche et Coordination Acoustique/Musique (IRCAM) in Paris. It had been highly regarded for its control over room geometry, absorption, and diffuse surfaces (Hartmann, 1983). The majority of

these components are mechanically set up and the acoustics are manually configured through motorized systems. However, there are disadvantages to many of these systems because the mechanics within them may generate a significant amount of noise. As introduced earlier, some digital systems provide acoustical compensation through multi-channel speakers. However, they do not automatically operate in real-time based on the dynamic content of the music performance.

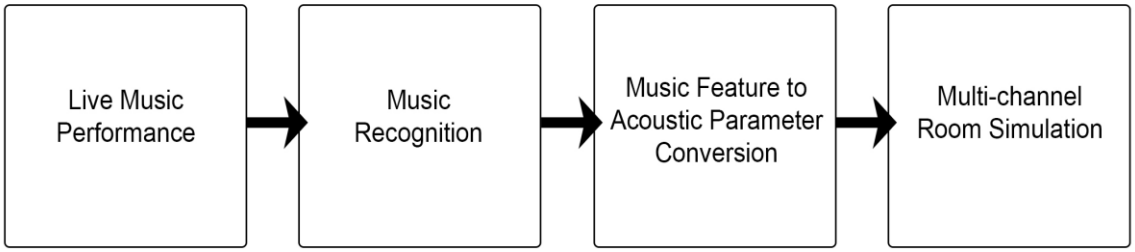

Figure 1.3: Basic concept of the room-adaptive system.

The room-adaptive system developed in this research is a mechanism that changes the acoustical characteristics of the room through an electro-acoustic system according to the content of music performed in the space. The analysis and processing occurs in real-time, instantly providing the appropriate room acoustics required for the type of music performed. Therefore, this system is intended to function in a fairly dry space through an array of speakers sufficient for simulating room characteristics.

The system takes an audio signal from a live source (i.e., a microphone, pickup, or synthesized sound) and sends the signal to multiple music recognition algorithms. The algorithms extract certain features of the music (such as tempo and loudness) and process them to become room acoustics parameters suitable for their corresponding musical characteristics. Then the acoustical parameters trigger the settings of the reverberation mechanisms, to simulate a room in real-time.

An application of this system may be in the use of multi-purpose halls that require variable acoustics. Unlike existing variable acoustics mechanisms, however, this system operates automatically and changes the parameter settings during a performance.

2. STRUCTURE OF SYSTEM

2.1 Platform and Signal Flow

The basic signal flow consists of the following steps: music recognition, parameter conversion, and multi-channel output. When the live music is performed in a space, the audio signal is captured through a microphone or is transmitted directly through an audio cable (if it is generated by an electric instrument). The signal is sent to a pre-amplifer, then to an interface, where it is converted from an analog to a digital signal. The converted signal is then sent to the computer to the Max/MSP patch. The Max/MSP patch consists of several music recognition algorithms that process the incoming audio signal in real-time to extract musical features. The extracted musical features are numerically converted to acoustical parameters, which are then automated to be entered into the input arguments of the room simulation (reverberation) modules that output the processed sound through a multi-channel speaker setup.

2.2 Approach in Matlab

Initially, the music recognition and real-time processing were attempted in Matlab, because some of the music retrieval algorithms found at the early stage of the research were Matlab codes and toolboxes. The preliminary idea was to analyze the sound signal coming from the interface and extract the salient musical features for each segment of a set time frame. The processed data would then be sent to Max/MSP via Open Sound Control (OSC) whereby the sound would be output through existing multi-channel reverb patches. For music recognition, University of Jyväskylä's "MIR Toolbox" was used. The code was set up in a way that Matlab would record, process, and send data to Max/MSP simultaneously in real-time. Although the code was successful in theory, there were many drawbacks and glitches that caused operation in Matlab to be ineffective and dysfunctional.

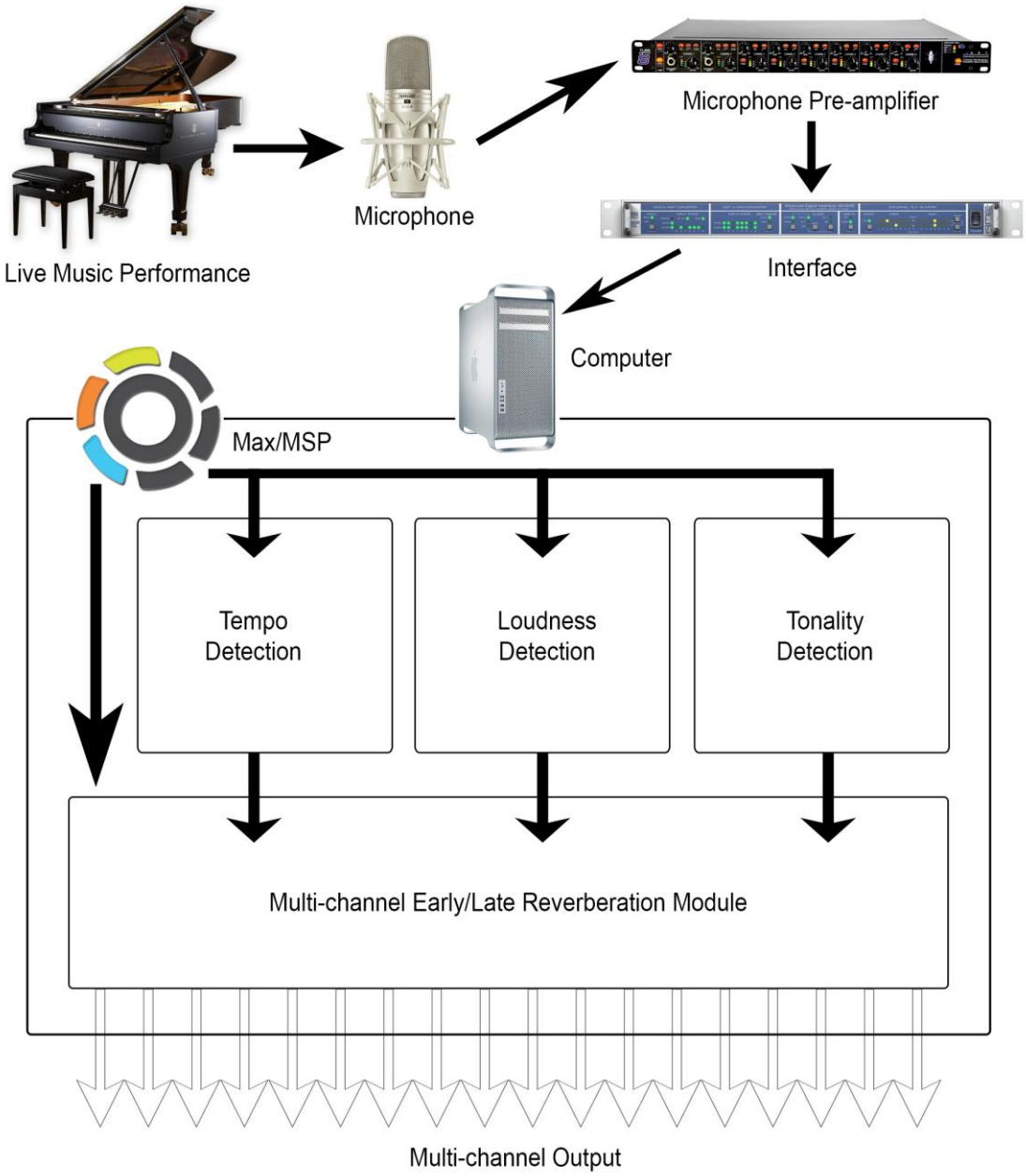

Figure 2.1: Signal flow chart of the system.

A key inconvenience in Matlab was the amount of delay, inevitable due to Matlab's method of "real-time processing." Matlab records the incoming audio signal from the computer's audio device, converts it into a data vector, and then analyzes the recorded data. Though the code had been set up to record and process the previously recorded data simultaneously, the time gap between the beginning of the recording and the beginning of analysis had to be set to a minimum of four seconds to analyze a data block of ten seconds. Though a latency of this length in processing is not crucial to the purposes of musical analysis, it begins to defy the concept of real-time signal processing. While Max/MSP technically operates using the same procedure, its processing is more efficient and causes a delay of only a few milliseconds. Max/MSP also does not require the incoming audio signal to be separated into increments of time for analysis. This inefficient processing in Matlab is a possibility that may have caused the program to crash at random instances, even when working with computers that have powerful processing capabilities.

Another factor to consider was the sophisticated routing between Matlab and Max/MSP. Though transferring data between the two software was possible through OSC, a disadvantage was that Matlab was unable to send and receive data simultaneously. Therefore, two computers were required to accomplish this task and furthermore, Matlab in the two computers had to be linked as well. In addition, the OSC data routing caused the instability in Matlab to become more severe. Therefore, the combination of the functions and for-loops in Matlab that continuously recorded, processed, and routed the data in real-time proved to be an inefficient and unsuccessful approach for the purpose of this thesis work.

2.3 Max Patch Setup

The entire process of music recognition, acoustic parameter calculation, and multi-channel output processing occurs in the digital domain in a single Max/MSP patch. In the patch, there is an audio player ("demosound" object) at the beginning of the signal chain, and it is capable of playing back the live input signal after analog to digital conversion ("adc~") or a pre-recorded "wav" file (see Figure 2.2). There is a gain control

to adjust the level of the signal in the output of the audio player. The signal is then routed from the audio player to multiple music recognition algorithms within the patch, as well as to the speaker output, after digital to analog conversion ("dac~"), in order to monitor the dry signal.

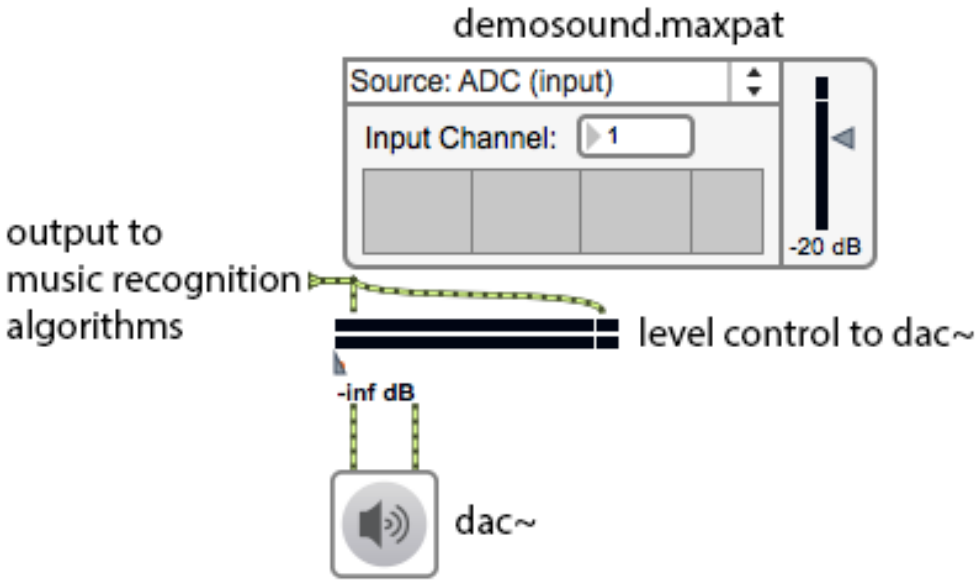

Figure 2.2: The beginning of the signal chain, starting with "demosound.maxpat". The dry signal is then sent to the music recognition algorithms and to the speakers ("dac~"). The level to "dac~" is turned down in the regular operation of this system, unless the dry signal needs to be monitored.

2.4 Music Recognition

As previously mentioned, the music recognition procedures combine several pre-existing algorithms to a single Max/MSP patch. The musical features that are extracted are tempo, loudness, and tonality. Each feature is a separate Max/MSP external developed by researchers in the fields of audio recognition. Numerous different algorithms have been tested, and those that had reasonable success for its purpose are used in this system. Other algorithms proved to be ineffective due to lack of accuracy, intermittent operation, and/or crashing errors.

2.4.1 Tempo

A tempo is a numerical value that represents the beats per minute for a given piece of music. In order to extract tempo from a music performance, a beat-tracker is used to detect the metrical pulse of the music. For the purpose of this thesis, Adam Stark's Max/MSP patch "btrack~" is used as the beat-tracker (Stark, n.d.), as it proved to be extremely accurate in its function.

Stark's beat-tracker is based on Daniel Ellis' programming algorithm (Ellis, 2007) and the tempo estimation method of Davies and Plumbley (Davies & Plumbey, 2007). The beat-tracker uses the onset detection function (DF), a representation that emphasizes note onsets by exhibiting peaks at likely onset locations. A continuous detection function is used to increase the accuracy in onset detection and to avoid missing appropriate onsets. The function needs to be sensitive to both percussive onsets, such as drum transients, and smoother onsets, such as a bowed violin (Davies & Plumbey, 2007). In order to achieve this, the *complex spectral difference* onset detection function is used instead of other different onset detection algorithms (Bello et al., 2005). This function works using both the frequency magnitude spectrum and the phase spectrum of the signal. The note onsets are emphasized through changes in energy in the magnitude spectrum, and/or changes in pitch (affected by the deviation in phase values) in the phase spectrum. Thus, this detection mechanism works both rhythmically and by pitch (Davies & Plumbey, 2007). Further details on detection of pitch changes through phase irregularities can be found in (Charpentier, 1986; Green & Patterson, 1969).

To calculate the detection function $\Gamma(m)$ at sample m, the Euclidean distance between the observed spectral frame $S_k(m)$ and the predicted spectral frame $\hat{S}_k(m)$ is summed for all frequency bins k,

$$\Gamma(m) = \sum_{k=1}^{K} \left| S_k(m) - \hat{S}_k(m) \right|^2. \tag{2.1}$$

For a derivation of this equation see (Bello, Duxbury, Davies, & Sandler, 2004). This method not only analyzes the audio signal that reaches the beat tracker's input, but also predicts the expected data. The detection function must have a fixed time resolution in order for the beat tracker to function with any sampling frequency of the input signal.

This time resolution is set to 11.6 ms per detection function sample, which was used in the onset detection work of Bello's work (Bello, Daudet, Abdallah, Duxbury, Davies, & Sandler, 2005).

Using this theory, Stark developed the "btrack~" patch (Stark, n.d.). However, this patch is only a beat-tracker; it identifies onsets and sends bang messages through its output. The output of "btrack~" is connected to Fredrik Olofsson's "f0.average_tempo" object (Olofsson, n.d.), which is a patch that calculates the average tempo from the incoming bang messages. The two codes together thus enable a real-time analysis of tempo from a music signal through beat-tracking.

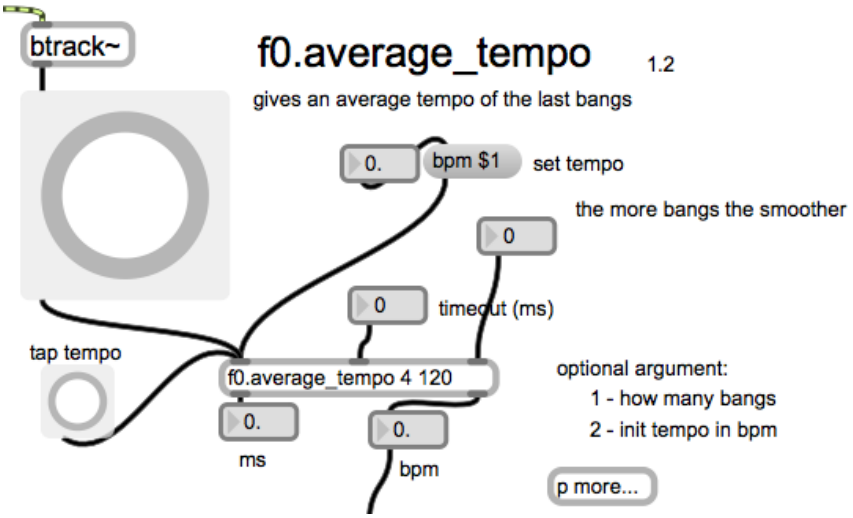

Figure 2.3: Tempo detection algorithm in Max/MSP, a combination of Adam Stark's "btrack~" and Fredrik Olofsson's "f0.average_tempo" object.

2.4.2 Loudness

Loudness is the subjective measure of the perceived intensity of sound. Studies by Fletcher and Munson compare loudness and sound pressure levels because human hearing is more sensitive at certain frequencies than others. Consequently, a relationship between the loudness and the sound pressure levels has been established as an ISO standard.

The equal-loudness-level contour (Figure 2.4) was generated with experimental data averaging measurements of subjects between ages of 18 to 25 years (ISO 226:2003,

2003). Then the loudness level L_N was derived based on the measurements, from the sound pressure level L_p at frequency f:

$$L_N = (40 \cdot \log B_f) \text{ phon} + 94 \text{ phon} \tag{2.2}$$

where

$$B_f = \left[0.4 \cdot 10^{\left(\frac{L_p + L_U}{10} - 9\right)}\right]^{a_f} - \left[0.4 \cdot 10^{\left(\frac{T_f + L_U}{10} - 9\right)}\right]^{a_f} + 0.005076 \tag{2.3}$$

T_f is the threshold of hearing, a_f is the exponent for loudness perception, and L_U is the magnitude of the linear transfer function normalized at 1 kHz.

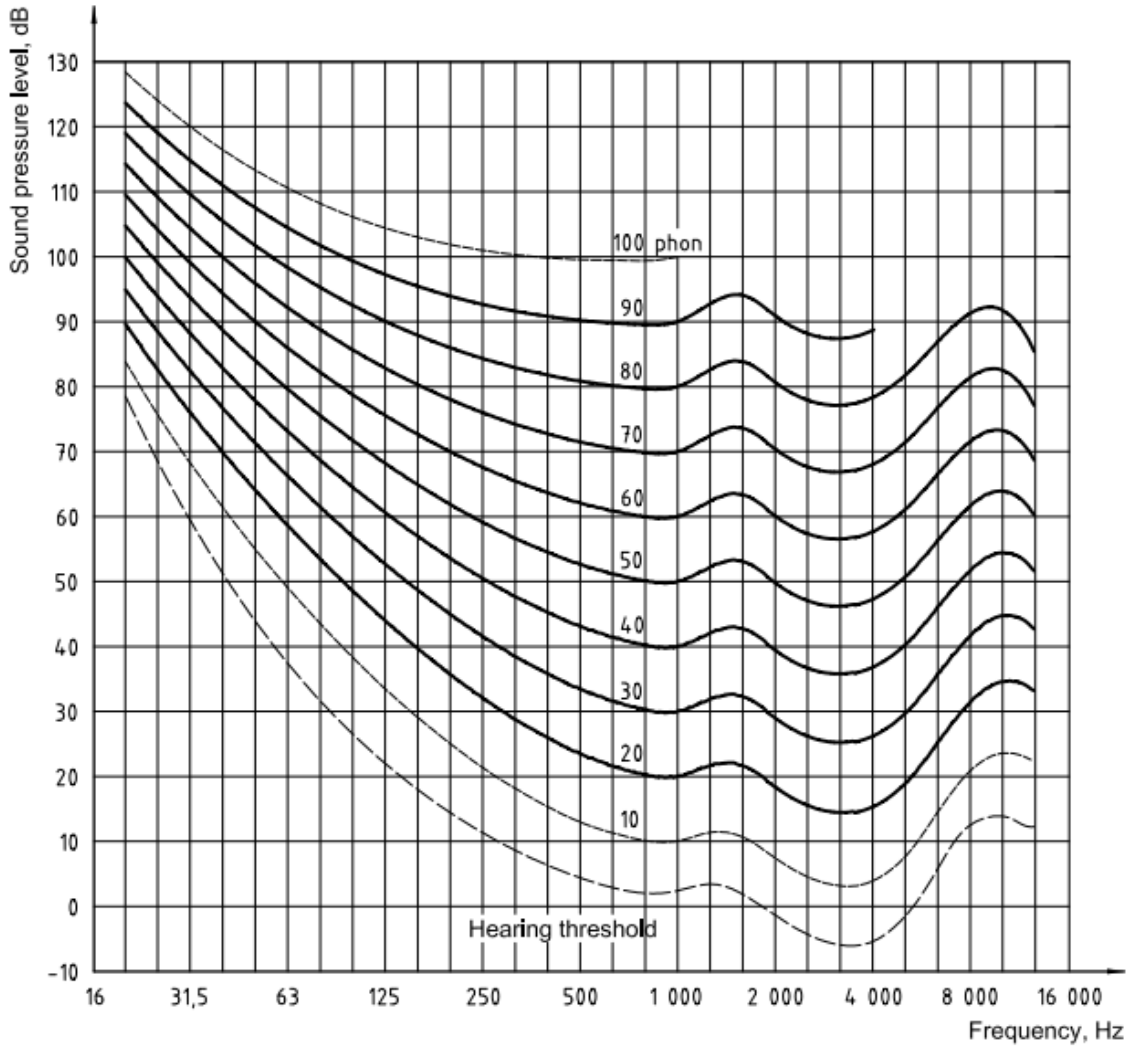

Figure 2.4: Equal loudness contour levels. Image taken from (ISO 226:2003, 2003).

The equal loudness contour levels, however, are only a presentation of loudness for sinusoidal pure tones. Because the audio signal analyzed is a musical performance, it consists of a broad range of frequencies, and thus loudness needs to be integrated over multiple critical bands. The total loudness of such complex tones is the sum of the loudness of the sound energy distributed and filtered in each critical band (Fastl & Zwicker, 2007). This results in a greater loudness than that of a single tone, even if both sounds had the same sound pressure level.

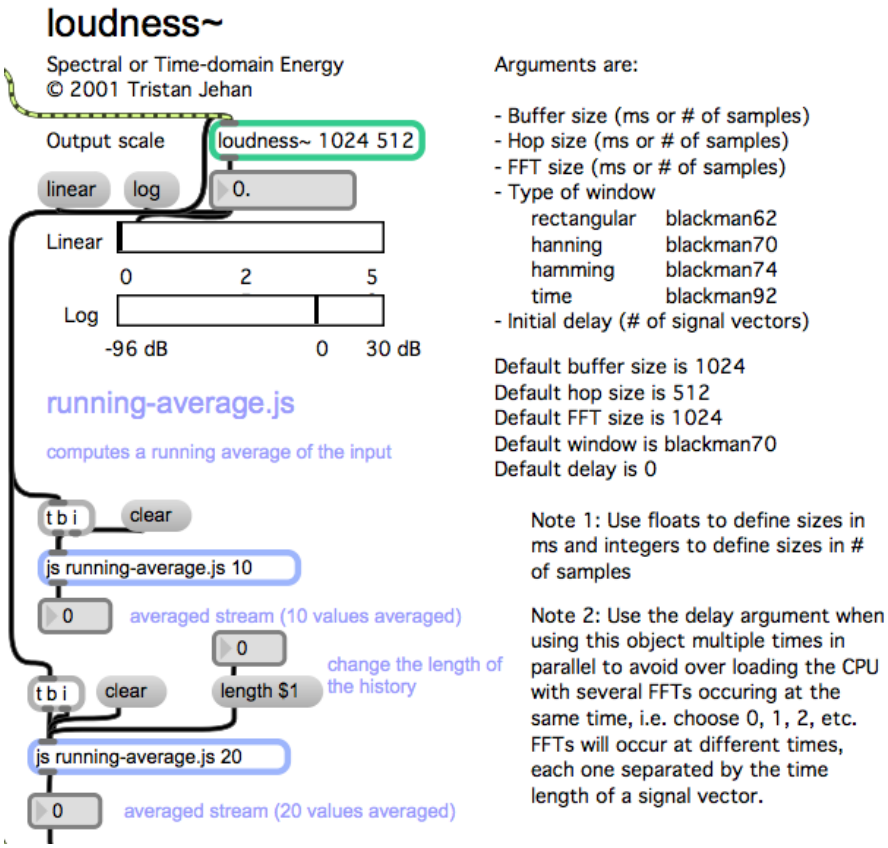

Figure 2.5: Tristan Jehan's "loudness~" function with its output averaged with John MacCallum's "running-average.js" function.

Tristan Jehan's "loudness~" Max/MSP patch is used for the extraction of loudness (Jehan, n.d.). The patch calculates spectral or time-domain power energy and outputs it in either linear or logarithmic scale, whichever is selected. For the purpose of the operation of this system, the logarithmic output scale is selected. The outputted value is

then averaged using John MacCallum's "running-average.js" object (MacCallum, n.d.). The object calculates the running average of the input signal, and an input argument is entered to specify over how many integer values the signal is averaged. The output of the object then displays the averaged stream of values. For the loudness, 20 values were averaged to provide output values smooth enough to send to the room parameters. The averaged levels provide a better sense of loudness than the instantaneous levels because signals of different durations have different perceptions of loudness. A signal of short duration, less than approximately 250-ms is perceived to be quieter than a signal of equal power but of longer duration (Yost, 2000). Thus, without the averaging process, the fast transients would cause the loudness levels to be overestimated.

Unfortunately, the detailed theory behind the operation of Jehan's code is unavailable. However, in his dissertation he mentions an equation for calculating loudness by adding amplitudes in the auditory spectrogram across all frequency bands:

$$L_{dB}(t) = \frac{\sum_{k=1}^{N} E_k(t)}{N} \tag{2.4}$$

where E_k is the amplitude of the frequency band k, of the total N (Jehan, 2005). Although further details on the methods for Jehan's code is unknown, it is tested to be successfully functioning when compared to some of the valid loudness calculating functions in Matlab.

2.4.3 Tonality

The algorithm used for tonality detection is Adam Stark's "chorddetect~", a Max/MSP object that detects chords in real-time (Stark, n.d.). A chord is any combination of two or more musical notes occurring simultaneously. A chord consists of two parts – the letter name and the quality. The letter name is determined by the root note of the chord, and the quality of the chord is classified by the combination pattern of the notes of the chord. Unlike many other chord-detection algorithms, this object is capable of detecting nine different chord qualities (Stark, Davies, & Plumbley, 2009). However, since the quality of standard tonalities is limited to major and minor keys, the chord qualities that the algorithm detects are modified to round to either major or minor.

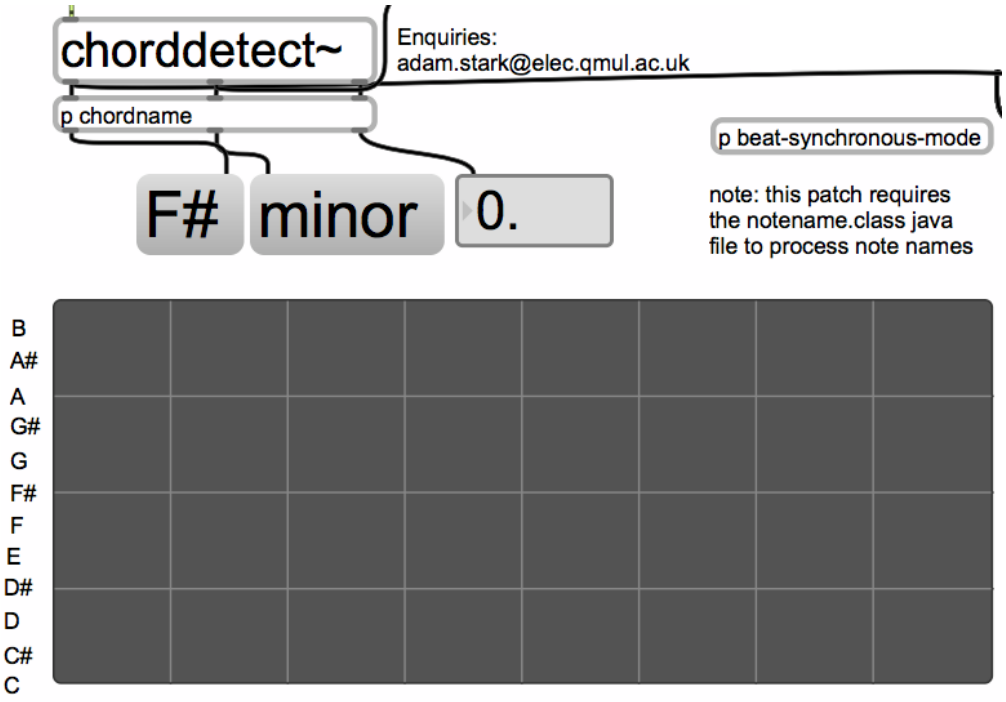

Figure 2.6: Adam Stark's "chorddetect~" function.

The theory within this chord detection algorithm is similar to other algorithms, with slightly different calculation methods. First, the audio signal is converted to a *chromagram*. A chromagram, also known as the pitch class profile, is a 12 x 1 vector that presents the amount of energy at each semitone's pitch class. Since there are twelve semitones that a musical note can be classified by, the chromagram plots the energy in relation to the nearest semitone that its frequency is associated with.

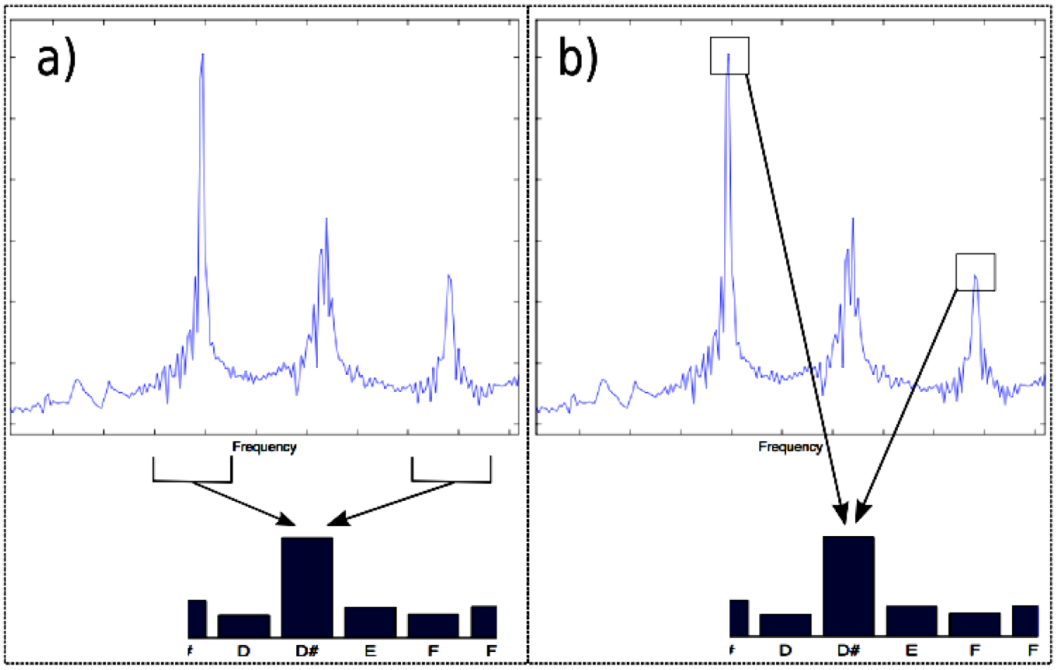

Figure 2.7: a) The bin mapping technique and b) the technique used in the "chorddetect~" algorithm, taking the energy peaks within the bin. Image taken from (Stark, Davies, & Plumbley, 2009).

The established chromagram is then compared to bit masks. A bit mask is a vector similar to a chromagram, except that the energy is rounded to be presented as either 1 (note present) or 0 (note absent). Pre-set bit masks exist as a database of chord profiles within the algorithm. The calculated chromagram is converted to a bit mask to then be compared to the chord profiles within the system. Once a match in the pattern has been detected, the data is then classified as a chord. One major difference between this algorithm and others is the method by which the chromagram is converted to the bit mask. Whereas most other pitch detection algorithms use the bin mapping technique in taking all of the energy within the spectral bin to assign to a specific pitch class, this algorithm takes only the energy peaks within the bin (see Figure 2.7). This increases accuracy in reducing unwanted energy (such as noise) within the procedure of pitch class mapping (Stark, Davies, & Plumbley, 2009).

2.5 Room Parameters

One of the key functions of this system is the alteration of room acoustics based on the musical performance. Thus, the conversion stage from the extracted musical features to the acoustical parameters is a crucial factor for the functionality of the system.

2.5.1 Reverberation Time

Perhaps the most significant room parameter affected by the music is the reverberation time. This system is set up so that the extracted tempo value affects the reverberation time. Conventionally, the reverberation time decreases as music becomes faster and the tempo increases (Goldsmith, 1944). Although this principle varies with the genre of music, at this stage of development the system has been set so that the reverberation time is only affected by the tempo, to avoid entering complications with music genre recognition.

In Max/MSP, the calculated tempo (output of "f0.average_tempo") is linearly scaled so that a tempo range of 60 – 240 beats per minute (bpm) converts into a reverberation time of 3 – 0.5 seconds. Thus, a tempo of 60 bpm would result in a reverberation time of 3 seconds, and a tempo of 240 bpm would be 0.5 seconds. The settings for this conversion can easily be modified, as the minimum and maximum values of tempo and reverberation time are the input arguments for the "scale" function used in the calculation. Therefore, for example, if the tempo of the music is 50 bpm (outside of the range of 60 – 240 bpm), and if the desired reverberation time is longer than 3 seconds (also exceeding the range of 3 – 0.5 sec.), the input arguments can be changed for the specific situation. However, the system's default is set to the range of the initially mentioned tempo and reverberation time, since most music falls within this range.

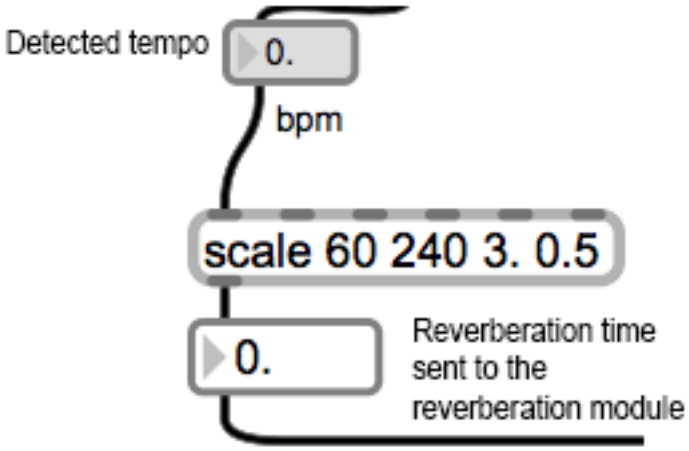

Figure 2.8: The tempo scaled and converted into reverberation time.

2.5.2 Direct to Reverberant Ratio

Another acoustical parameter influenced by the musical performance is the direct to reverberant ratio. Direct to reverberant ratio is the relationship between the intensities of the direct sound and the reverberant sound. In this system, loudness of the music determines the direct to reverberant ratio. Studies have shown that a sound is perceived to be more reverberant if it is perceived to be louder (Lee & Cabrera, 2010). Thus, in this system, with more loudness reverberation is increased.

The output from Jehan's loudness algorithm, once averaged, is sent to a scale function. Because the logarithmic output of the "loudness~" function ranged from -96 – 30 dB, the input argument values for the input range of the scale is set to -96 and 30. On the reverberation module, there is a "dry/wet" slider that controls the direct to reverberant ratio. Its input arguments range from 0 to 127, with 0 being the driest and 127 being the wettest. The output range of the scale function is set to 127 and 50, so as the signal becomes louder, it becomes more reverberant. 50 is an appropriate minimum wet level in order for the quieter sounds to not be completely dry. This setting, similar to that of the tempo, can also be altered if necessary. The automated dry/wet slider thus

becomes a volume slider for the output from the reverberation module. Further details on the output signal flow is discussed Section 2.6.

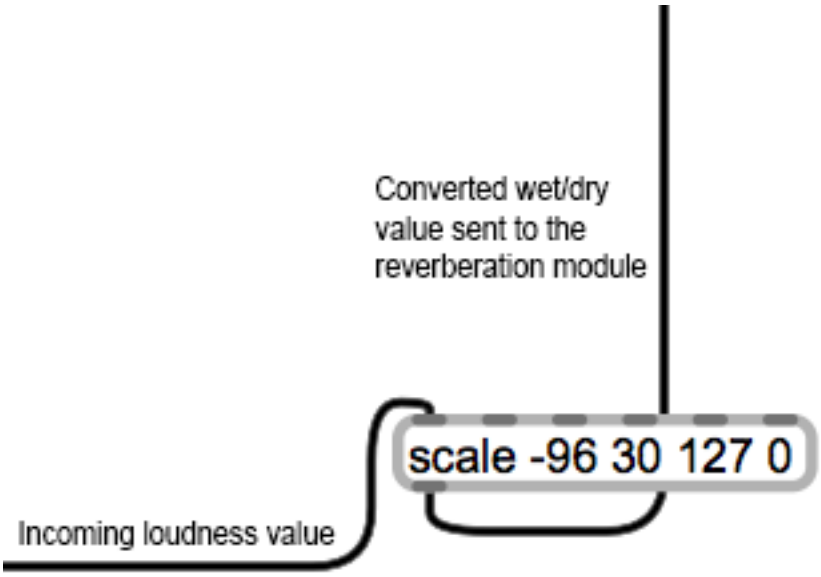

Figure 2.9: The averaged loudness scaled and converted to wet/dry value.

2.5.3 Low Pass Filter from Tonality

Music can be perceived to be either brighter or darker, depending on the key of the music. A *key* in music is defined by the tonal center's pitch, eliciting a sense of arrival and rest. Other accompanying notes and/or chords that create varying degrees of tension resolve upon the return to the tonic. The presence of a key defines the tonality; music without a key would be considered atonal. The majority of music is tonal and has a key associated to it. The system developed in this thesis is designed to work with tonal music, because its changes in brightness are controlled by the detected key.

Keys are classified with sharp and flat symbols. The C major and A minor keys are the only keys that do not have any sharps or flats. As sharps are added, the key is incremented by an interval of perfect fifth, starting at the key of C. As flats are added, the key changes downwards by intervals of perfect fourths. Figure 2.10 visually represents this theory in what is known as the *circle of fifths*. Further in-depth explanations of this may be found in the music theory literature.

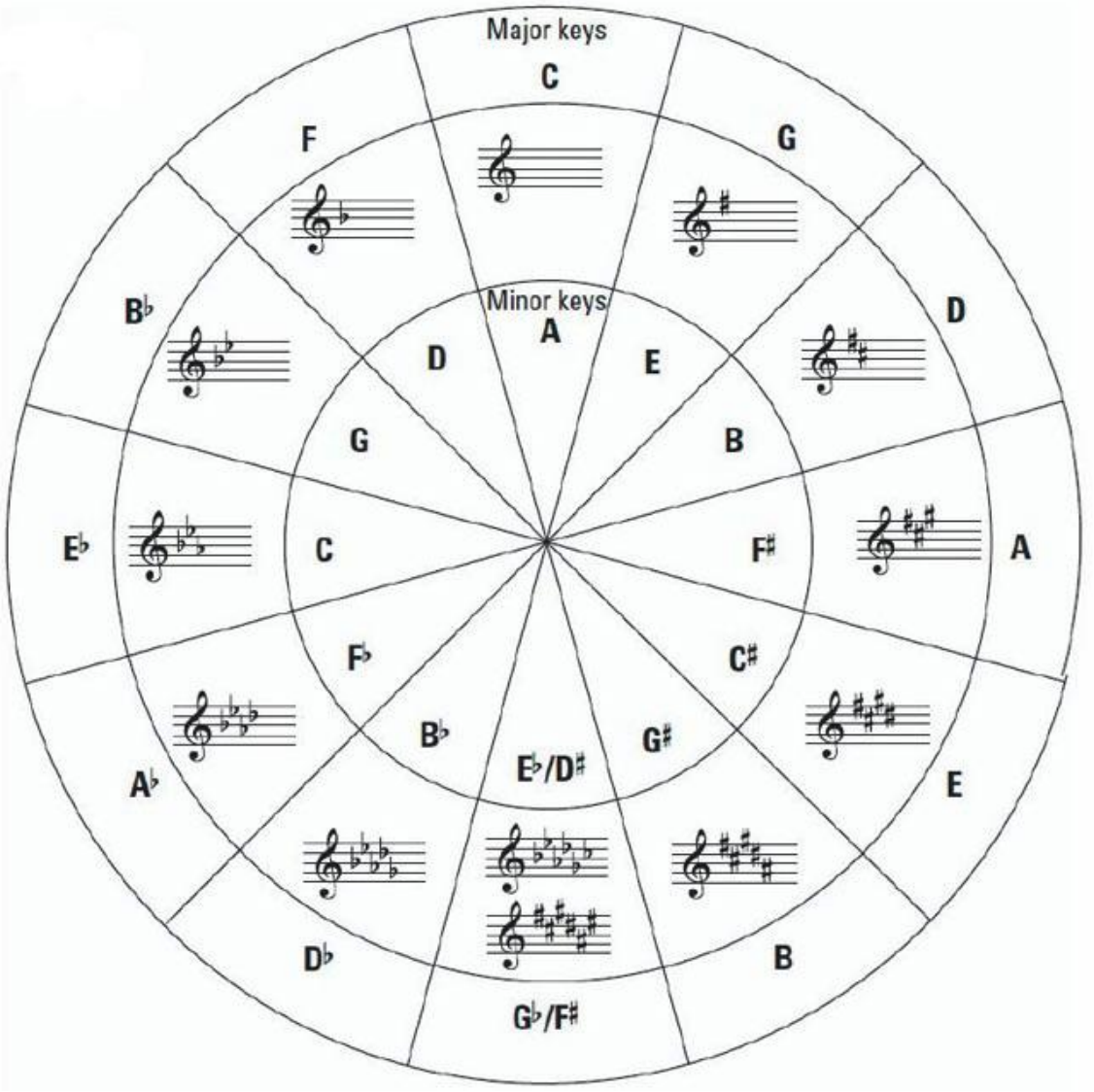

Figure 2.10: The circle of fifths. Image taken from (Chappell, 2013).

Over the course of the time, many different tuning systems have been developed. Almost all tuning systems use the interval of an *octave*, a frequency ratio of 2:1. Two notes an octave apart are heard by the human ear to be essentially the same, due to their closely related harmonics. Most modern Western music today divides the octave equally into twelve semi-tones. However, the octave has not always been divided this way; different tuning systems each had different methods of dividing the twelve semitones.

For music based on harmony (containing notes that are occurring simultaneously), the notes in a chord must sound in tune in relation to each other to be pleasing to the ear. This is determined by the relationship between the frequencies of the notes and their harmonics within the tuning system.

One of the earlier tuning systems was the just intonation system (Stoess, n.d.). It tuned the important major triads to be pure, with the ratio of 4:5:6. The Pythagorean tuning system, using the concept of the just intonation system, was based on Pythagoras' mathematical system that defined the ratio of the interval of fifths to be 3:2, and the fourth being 4:3. The ratio for the interval of the major second (whole step) was calculated to be 9:8. Using this approach, the rest of the octave was divided and tuned with the appropriate ratios. The third interval contained an impure ratio and was thus considered dissonant at the time. The Pythagorean tuning system also contained the wolf interval, which was a single fifth interval (E^b and G#) that was smaller than the eleven other fifths. This was an especially dissonant interval that had a sound of a wolf's howl, which is what gave it its name.

The mean-tone system, developed during the Reformation period, was based on tuning the major thirds to be pure (Stoess, n.d.). This resulted in the fifths becoming slightly flattened. However, the fifths were flattened in varying degrees over four fifths, to reduce dissonances and make the fifth interval more acceptable. Then the well-temperament was developed. It began to be used in the time of Bach and Handel. This tuning system distributed the pure and smaller fifths more evenly, making the dissonance less perceivable. As a result, the well-temperament system did not contain the wolf-fifth interval and also enabled compositions to be played using all keys and chords. With this tuning system, each key had a different perception of harmonic brightness. Keys sounded brighter with the increasing number of sharps, and darker with more flats. Thus, in the circle of fifths, the keys became brighter as they progressed clockwise and darker as they progressed counter-clockwise.

This sense of harmonic brightness and darkness is emphasized through the room acoustics by applying a low pass filter. First, the detected chord from the "chorddetect~" algorithm is converted to either a major or a minor chord (as mentioned in Section 2.4.3). Then the resulting chord is sent through a series of "if" functions, outputting the

appropriate value for brightness level. There are seven brightness levels (0 – 6) and they are averaged over 10 values using the "running-average.js" object, as the chord detection algorithm reacts too rapidly for the low pass filter to function realistically. As the extracted tonality is detected to be darker, the cutoff-frequency for the low pass filter of the reverberation decreases. This again is achieved using the scale function, with the cutoff frequency of the low pass filter ranging from 10 kHz (dark) to 20 kHz (bright). The resulting output of the scale function (cutoff frequency value) is then sent to a line function so that the cutoff frequency will not leap from one value to another, but instead will smoothly increase or decrease. The output of the line function is then connected to the cutoff frequency input argument of the low pass filter in the reverberation module. The low pass filter is applied within the feedback delay network rather than the dry input signal, and thus results in different reverberation times across frequency.

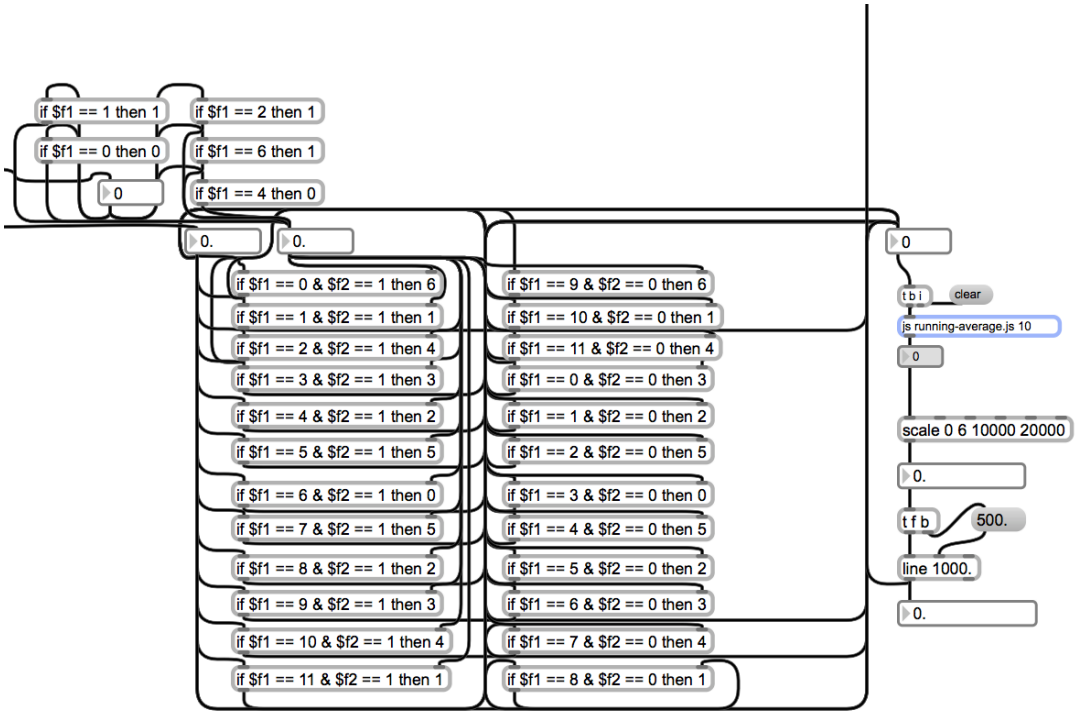

Figure 2.11: The "if" functions receive chord information in numerical values (letter name and quality separately), and convert them to a brightness value. Then they are averaged so that the value does not change too rapidly. The averaged value is then scaled and converted to a cutoff frequency. The cutoff frequency is routed to the reverberation module through a line function, so that the changes are gradual.

2.6 Multi-Channel Output

For the output of this system, a multi-channel output is necessary to sufficiently simulate room acoustics. Two reverberation modules are used – the Virtual Microphone Control (ViMiC) system (Peters, n.d.) and Nils Peters' FDN Late Reverb (Braasch, Matthews, & Peters, 2010). The ViMiC system simulates and outputs the early reflections while Peters' FDN Late Reverb outputs the late reflections. Because the late reverberation does not contain directional information, it operates using the concept of a feedback delay network (FDN). The FDN uses multiple delay lines through a Hadamard mixing matrix, dispersing the energy of every input to every output. Though the number of channels can be altered, this system is primarily designed to output into 16 channels.

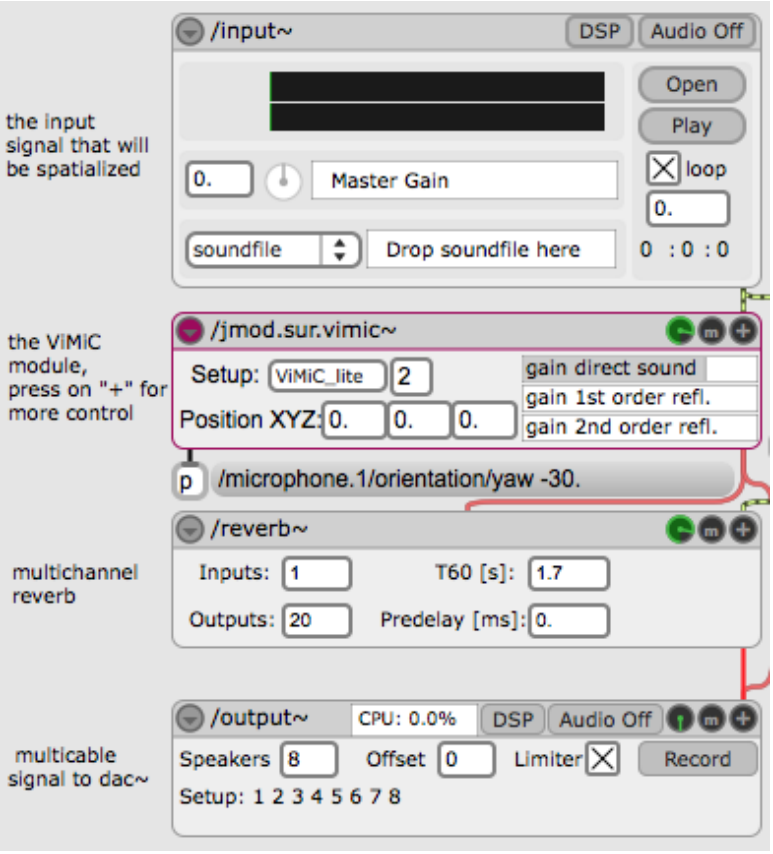

Figure 2.12: ViMiC parameters in Max/MSP ("mod.sur.vimic~").

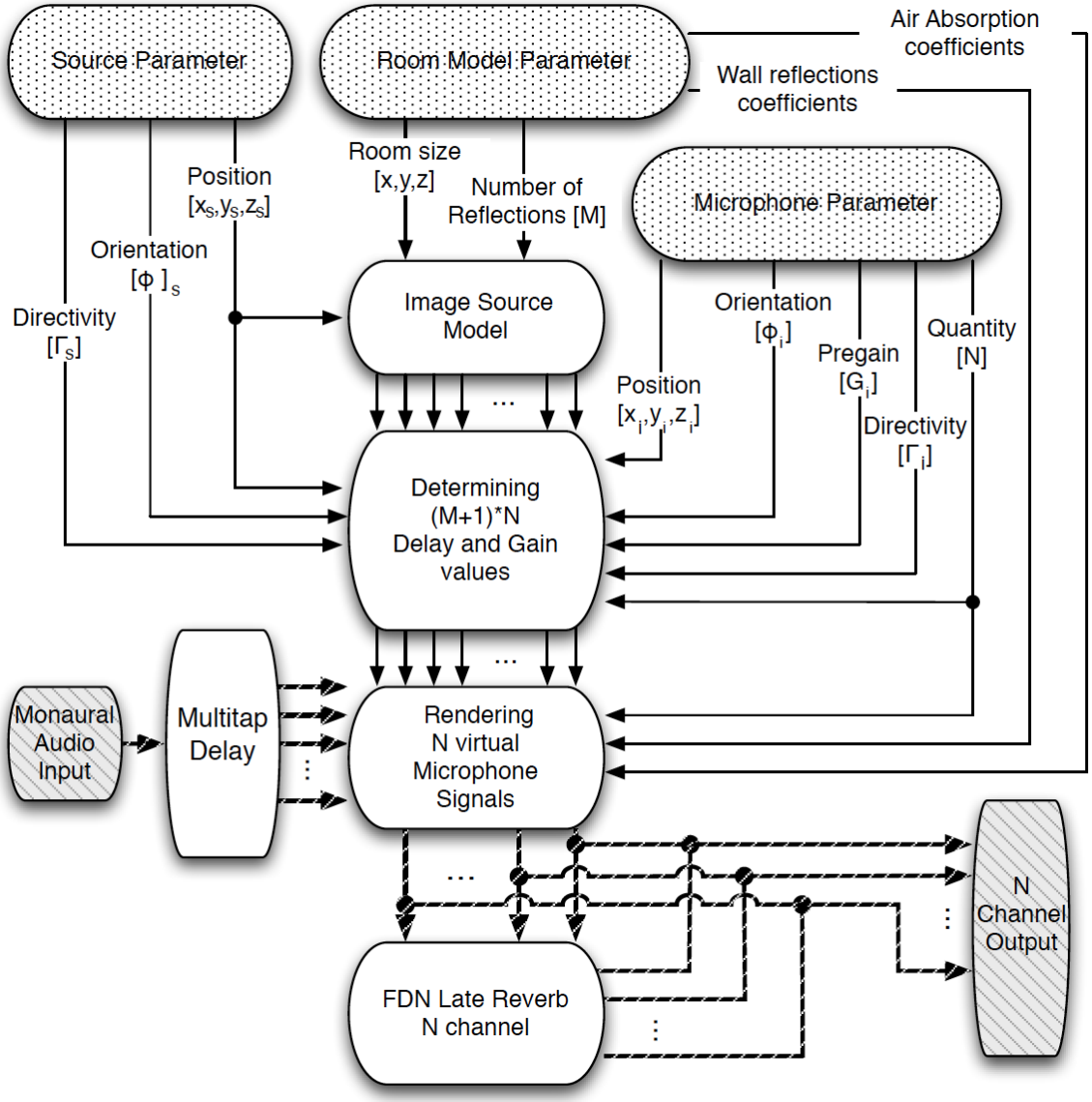

Figure 2.13: Flowchart of the reverberation module. Image taken from (Braasch, Matthews, & Peters, 2010).

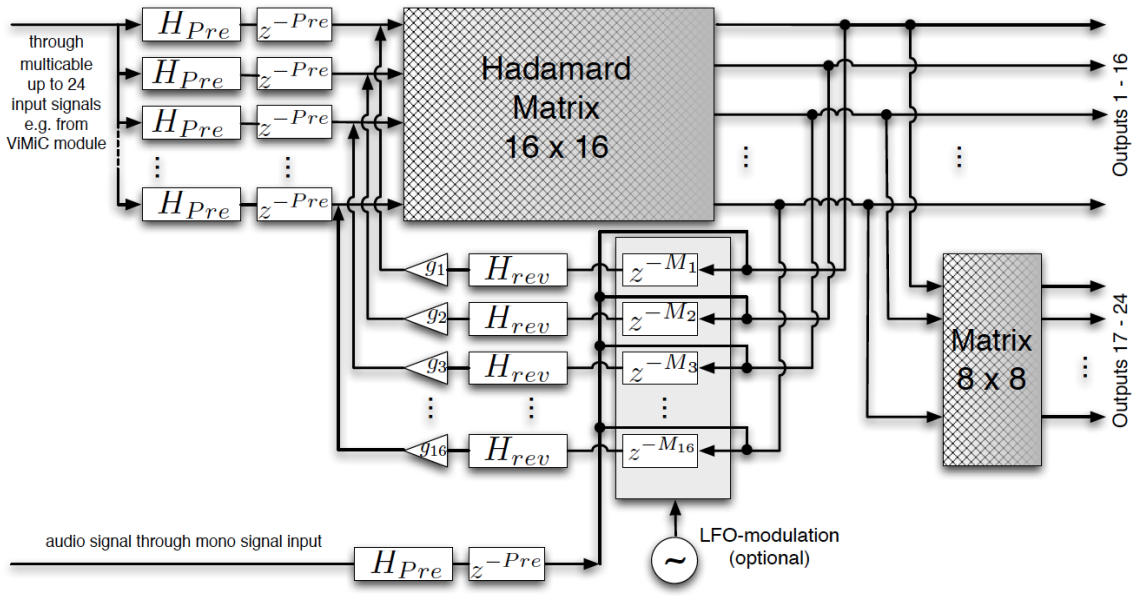

Figure 2.14: Flowchart of the Nils Peters' FDN Late Reverb module. Image taken from (Braasch, Matthews, & Peters, 2010).

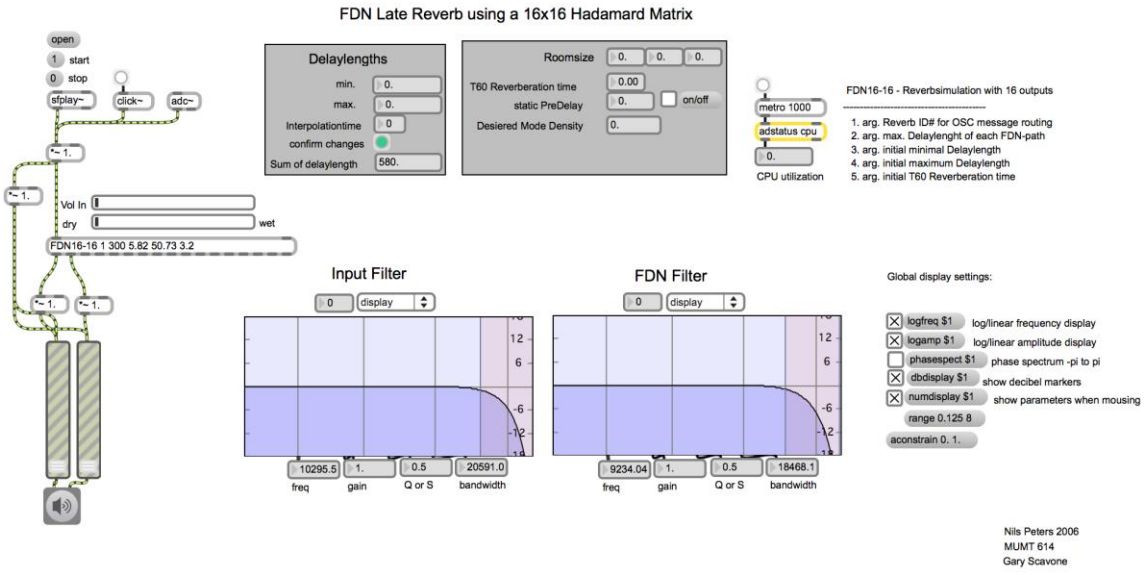

Figure 2.15: Nil Peters' FDN Late Reverb module in Max/MSP.

In the Max/MSP patch, each of the previously mentioned musical features feeds into the appropriate input argument of the reverberation module that corresponds to the associated room acoustics parameter. As it is the purpose of this system, the three parameters relating to a musical feature are automated. However, the settings for automation can be flexibly changed, as previously explained in this paper.

The multi-channel speaker setup for this system is not intended to be used as a sound reinforcement system amplifying the dry signal, but to only alter and simulate the acoustics in the space. Therefore, as mentioned in Section 2.5.2, the input for the dry signal in the reverberation module is disconnected. As a result, the "wet/dry" slider (refer to Figure 2.15) becomes a volume slider for the reverberation. If there is a need to amplify the dry signal through the multi-channel speakers, the dry signal can be fed separately to whichever speaker channels desired, in addition to the reverberated signals.

3. TESTING OF THE SYSTEM

After the music recognition algorithms was put together and set up to work with the reverberation modules for room simulation, the system was tested. Since the settings for the acoustical parameters can be changed to suit different environments, the focus of the testing was on the recognition functionality.

It is to be noted that many of the subjective results were evaluated through the author's introspective approach rather than conducting psychophysical experiments with test subjects. The author is an expert in the field of music and sound engineering, and also had formal training from Berklee College of Music.

3.1 Accuracy of Recognition

As music recognition plays a significant role in this system, testing the accuracy of the recognition algorithms was necessary. The system was tested with various types of music under different conditions.

3.1.1 Different Genres

The following genres were tested with each of the algorithms: pop, funk, classical, rock, jazz, a capella, bossa nova, and opera. The selected music varied in genre, tempo, key, and instrumentation, including both instrumental and vocal pieces. Existing compositions were played live by a mixture of professional, semi-professional, and amateur musicians. Excerpts of approximately 2 minutes were played for each piece, and each musician had an active metronome with a designated tempo for each genre except for classical, opera, and a capella. Detailed information on the music used for this test is in Table 3.1.

Table 3.1: List of musical pieces used for testing different genres.

Genre	Title of piece	Key	Tempo	Instrumentation
Pop	Sunday Morning (Maroon 5)	C	88	Vocalist, Guitar, Piano, Bass, Drum Set
Funk	Rio Funk (Lee Ritenour)	G	110	Guitar, Piano, Bass, Drum Set
Classical	Piano Sonata No. 11 (Wolfgang Amadeus Mozart)	A	Free	Piano
Rock	Don't Stop Believin' (Journey)	E	117	Vocalist, Guitar, Piano, Bass, Drum Set
Jazz	Autumn Leaves (Joseph Kosma)	Gm	210	Piano, Bass, Drum Set
A Capella (Jazz)	Silent Night (Franz Xaver Gruber)	A^b	Free	5 Vocalists
Bossa Nova	The Girl From Ipanema (Antônio Carlos Jobim)	D^b	120	Vocalist, Guitar, Piano, Bass, Drum Set
Opera	Lascia ch'io pianga (George Frideric Handel)	E	Free	Vocalist, Piano

All testing in this section was done in a rehearsal space without the multi-channel speaker setup. This is acceptable because only the music recognition process is tested, and the affected acoustical changes are not observed in this test. The feedback that may be cause by the acoustical changes to the music recognition is not considered in the tests, as it is assumed to be irrelevant in this system because the audio is either close miked or fed by a line signal.

There have been two approaches for the routing of the signal to the system. The first is placing an omni-directional microphone in the middle of the performers, capturing the music performance with the single microphone and sending it to Max/MSP. The second approach is placing microphones for vocalists and instrumentalists that do not have a direct line output. Then the individual signals from the microphones and/or instruments

are mixed and summed through a mixer. The mixed signal is then sent to Max/MSP as a mono signal.

The first approach has more of a “raw” sound where the amplitude balance of the instruments must be adjusted within the performance by the musicians. There is no separate amplification or post-processing (such as EQ or compression). It is more of a natural method and captures the sound of the music in a way that one would hear without any sound reinforcement systems. It is suitable for genres such as classical, opera, and sometimes jazz.

The second approach is for a more modern sound. It is for genres of music in settings where the instruments such as the drum set overpowers other instruments in amplitude and/or vocalists are inaudible. Generally, the signal has a greater signal-to-noise ratio because most instruments are individually close-miked. This approach is often necessary in genres such as pop, funk, and rock. Although post-processing can be applied to the audio to make the sound more aesthetically pleasing, only the levels of each track were balanced in order not to deviate too far from the first approach.

Table 3.2: Genre accuracy comparison results.

Genre	Tempo	Loudness	Tonality
Pop	Accurate	Accurate	Accurate
Funk	Accurate	Accurate	Inaccurate
Classical	Accurate	Accurate	Accurate
Rock	Accurate	Accurate	Accurate
Jazz	Accurate	Accurate	Inaccurate
A Capella	Inaccurate	Accurate	Inaccurate
Bossa Nova	Inaccurate	Accurate	Inaccurate
Opera	Inaccurate	Accurate	Accurate

Table 3.2 shows the results of the genre comparison test. The test was performed for two minutes per genre with observation on the reaction of the tempo, loudness, and tonality algorithms. The accuracy of the tempo was determined by the algorithm displaying a reasonable proximity to the tempo of the metronome, and for the pieces

without a metronome (free tempo), the visual bang output from the tempo algorithm was observed to determine whether it was roughly similar to the performance heard.

The tempo algorithm was fairly accurate for most types of music. For pop and rock, the detection was slightly unstable in the parts where percussive instruments such as drums were not playing. The tempo for jazz was also accurate despite its swing rhythm, excepting that the algorithm was detecting the tempo in half-time (half of the tempo). In jazz music, counting in half-time is natural; however, the tempo would still be twice as fast. Therefore, although the calculated tempo was incorrect, it is recorded as "correct" because the algorithm was still counting in time. Meanwhile, for the bossa nova sample, the tempo detection was inaccurate due to the rhythmic complexity (such as numerous syncopations and flams). The algorithm detected the up-beat notes as down-beats and attempted to establish a new tempo based on the up-beat rhythm. The algorithm was also unable to keep up with the opera piece because the piece was very slow yet free in tempo; each measure had drastic tempo variations that the algorithm was unable to predict.

The loudness algorithm was accurate in its detection for all genres. The extracted values from the output of the "loudness~" object were overly sensitive and drastic for the purpose of this system, especially with percussive transients; however, the averaging softened and smoothened the dynamic movement and also consequently correlated to the loudness that was heard by the author. As mentioned in Section 2.4.2, without the averaging process, the results would have been considered inaccurate because the ears do not sensitively detect loudness as the "loudness~" algorithm does.

The system's tonality algorithm operated least accurately. The results indicate that it has limitations and difficulties with extended polyphonies and sophisticated chords. Chords with tensions (non-diatonic notes) were detected to be incorrect most of the times, and thus genres such as jazz, bossa nova, funk, and a cappella that use these chords had inaccurate tonality detections.

3.1.2 Static vs. Dynamic Music

Some of the music repertoire mentioned in Section 3.1.1 was altered and performed differently to compare the recognition of different static and dynamic music. Some of

the results for this comparison have been described in the previous section; however, further testing was performed for further comparisons.

"Lascia ch'io pianga" by George Frederic Handel (the opera example in Section 3.1.1) was an ideal example for this test. This piece contains a wide range of dynamics, flexible tempo, and sufficient chord changes. In order to imitate this sort of variation, a jazz song and a pop song were performed with exaggerated dynamics (loudness) and tempo changes. A dynamic orchestral piece was also added to the repertoire for testing. The orchestral piece was an anechoic recording of Mozart's Overture from "The Marriage of Figaro" (Mozart, 1992), and was fed to the input of the Max/MSP system directly.

Meanwhile, for the static repertoire, music was performed in a one-chord, loop style. For this part, three different styles of music were played as well. The first was music consisting of a vocalist and a pianist. The pianist played quarter-note voicings of the E chord at a steady slow tempo, with no variation in dynamics. The vocalist sang improvised notes that were diatonic to the E key, in half-notes. This was the "static" version of the "Lascia ch'io pianga", similar to the original except that the tempo, dynamics, and chords were fixed. The rest of the static pieces were similar to this. For jazz and pop, the musicians were playing a vamp on a single chord in a steady tempo (fast for jazz, medium for pop). The fourth piece was created through the use of VST plugins. Using a combination of Kontakt VSL, LA Scoring Strings, and Session Strings Pro, a medium tempo orchestral piece containing simple legato eighth note voicings of a single chord was created. There was no reverberation added to the piece, and the recording was sent to Max/MSP in the same way that the Mozart piece was routed.

Table 3.3: Static vs. dynamic music comparison results.

	Genre	Tempo	Loudness	Tonality
Static	Opera	Inaccurate	Accurate	Accurate
	Pop	Accurate	Accurate	Accurate
	Jazz	Accurate	Accurate	Accurate
	Orchestral	Inaccurate	Accurate	Accurate
Dynamic	Opera	Inaccurate	Accurate	Accurate
	Pop	Accurate	Accurate	Accurate
	Jazz	Inaccurate	Accurate	Inaccurate
	Orchestral	Inaccurate	Accurate	Inaccurate

The results for the recognition of the dynamic music were not too different from that of the previous genre comparison accuracy test (see Table 3.2). There was an addition of orchestral music to the repertoire, and it was recognized inaccurately, due to its frequent changes in rhythm and busy harmony. Static music recognition resulted to be mostly accurate, except for the tempo recognition in opera and orchestral. Even if the tempo is static, the legato characteristic of the orchestral and opera music hindered the rhythmical consistency. In opera music, even if the piano played a straight rhythmic accompaniment, the sound of the powerful vocalist dominated the inputs of the algorithms. Therefore, this experiment reveals the tempo algorithm's ineffectiveness when distinctly detectable transients are not present.

3.1.3 Recorded vs. Live Audio

A comparison between pre-recorded music and a live music performance was also tested. There were two parts for this test. For the first part, anechoic audio was used for the pre-recorded music to better match the live audio signal. The purpose of this part was to observe the algorithms' reactions towards detecting a signal played by the computer (through the "sfplay~" function) in comparison to the live signal input from the "adc~" function.

A 30-second guitar track and a 15-second piano track were selected for the anechoic audio to be tested. Then, for the live performance, the anechoic music was transcribed and performed note for note. In comparing the algorithms' results, no differences in the recognition between the two signal sources were experienced.

The second part of this test was testing music that was recorded, mixed, and mastered. This was not compared with any live signal; its sole purpose was to observe the functionality of the detection algorithms with commercially recorded and distributed music. It is common to see musicians perform live with a pre-recorded track playing in the background; therefore, the recorded signal with a sonic quality of a market-released backing track was to be tested.

Five diverse pieces of music were chosen for the testing of the recorded music. They were loaded to the Max/MSP system as "wav" files and played back through the music player in the "demosound" object.

Table 3.4: Recorded audio recognition results.

Title	Genre	Tempo	Loudness	Tonality
Autumn Leaves (Kosma, 2003)	Jazz	Accurate	Accurate	Inaccurate
Billie Jean (Jackson, 1982)	Pop	Accurate	Accurate	Accurate
It's My Life (Bon Jovi, 2000)	Rock	Accurate	Inaccurate	Accurate
Symphony No.5, Movt.1 (Beethoven, 2005)	Orchestral	Accurate	Accurate	Accurate
Magic Flute (Mozart, 2005)	Opera	Inaccurate	Accurate	Inaccurate

The results for this test were similar to those of the live audio. This indicates that most of the audio processing such as compression, reverb, EQ will not have any effect on the recognition process. This also shows that taking the earlier mentioned approach in balancing the levels through a mixer before sending to Max/MSP can have post-

processing applied without affecting the recognition system. However, there is one exception to be noted. As indicated in Table 3.4, the loudness for the rock music was inaccurate. It was because the loudness for the song remained at a constant level, despite the wide range of dynamics that the song musically contained. This is assumed to be due to the heavy compression applied to the track. The compressed amplitude directly translated as a constant loudness level, although the author still perceived the music to be louder in some parts than others. Compression was audible in this song, but it was not at a level where it would destroy the perceived dynamics. The loudness algorithm, however, was unable to detect the loud and the quiet parts of the song. Some of the other recordings in this test also had compression over the entire mix (as most music that has been through the mastering process would), but to a lesser degree than the rock song, and thus did not seem to affect the loudness algorithm.

3.1.4 Single vs. Multiple Instruments

Recognition accuracy was also compared between music with a single instrument and music with multiple instruments. The single instruments for the testing were acoustic guitar, piano, and voice, whereas the instrumentation for multiple instrument testing was vocals, guitar, piano, bass, and the drum set. "Don't Stop Believin'" was chosen as the repertoire to be played (and sung) by the musicians in each of the four cases to be tested, to maintain consistency within the test. For the single instrument test, it was arranged accordingly for each instrument, and for the test with multiple instruments, the five musicians performed the piece together.

Table 3.5: Single vs. multiple instruments comparison results.

Instrumentation	Tempo	Loudness	Tonality
Guitar	Inaccurate	Accurate	Accurate
Piano	Accurate	Accurate	Accurate
Voice	Inaccurate	Accurate	Inaccurate
Voice, guitar, piano, bass, drum set	Accurate	Accurate	Accurate

The results in Table 3.5 show unexpected results. According to the algorithms' instructions the detection is expected to be more accurate with simpler music and a single instrument. However, the recognition algorithm was fully accurate only in analyzing the piano version and the version with multiple instruments. This may be because with the musical nature of solo performances, the guitar may not have a strong enough rhythmical base to trigger a tempo for the algorithm to detect, and the voice has neither the rhythmical nor the harmonic support in its content of performance that the tempo and tonality algorithms can successfully recognize. The piano performance, however, contained both the melody and the chord accompaniment with stable rhythmic support.

Therefore, to determine whether the affecting factor was the number of instruments, another test was done with modifications. The music was performed with two instruments rather than one. Three rounds of testing were done with the duo of voice with guitar, voice with piano, and guitar with piano. In the duets with voice, the voice sang the melody while the guitar or piano played the accompaniment for it. In the duet with piano and guitar, the guitar played the melody while the piano accompanied it. This resulted in all algorithms accurately detecting each of the parameters in all three performances.

Since all three parameters were detected accurately for both the fewer and multiple instrument conditions, the same test was done with a different repertoire. Referring back to Table 3.2, the bossa nova performance was not detected accurately when performed in a full band setting. Therefore, "The Girl from Ipanema", the same bossa nova piece, was chosen to be the music tested. As the previous test proved that the algorithm better recognized a duet due to its rhythmic and harmonic support, the solo performance tests were replaced by the duo performances. Thus the test was done with three duets and one full band performance.

Table 3.6: Results with bossa nova.

Instrumentation	Tempo	Loudness	Tonality
Voice, guitar	Inaccurate	Accurate	Inaccurate
Voice, piano	Inaccurate	Accurate	Inaccurate
Guitar, piano	Inaccurate	Accurate	Inaccurate
Voice, guitar, piano, bass, drum set	Inaccurate	Accurate	Inaccurate

Although the versions with the reduced instrumentation are much simpler in musical content than the full band version, the algorithm still detected the tempo and tonality inaccurately. The testing results indicate that although the number of instruments does have an effect the accuracy of the algorithm, the key factor in its accuracy is the content of the music itself. Simpler polyphony and a strong rhythmical foundation are necessary for the algorithm to operate accurately.

3.1.5 Dry vs. Reverberant Spaces

An element of interest is the environment in which the system is set up. It has been previously mentioned that the system operates ideally in an electro-acoustic space. The space must be relatively dry, in order for the artificial acoustics generated by the multi-channel output to have its full-intended effect. Though determining the reverberation threshold at which the system fails to function effectively would be desired, setting up a multi-channel output speaker system in various reverberant and dry spaces to test this is difficult. Therefore, in this section, the effect of the space's reverberation on the recognition process is tested instead.

The rehearsal space in which the tests of the previous sections of this chapter were performed is relatively dry, with a reverberation time of around 0.7 seconds. Therefore, tests were done in comparison to three other spaces with varying reverberation times to compare the recognition behaviors in reverberant spaces. Table 3.7 lists the tested spaces and their reverberation times. Each space's reverberation time was calculated using a measured impulse response.

Table 3.7: Tested spaces with different reverberation times.

Space	Reverberation time (T30)
Small rehearsal space	0.7 seconds
Small room	1.1 seconds
Dry church	1.7 seconds
Reverberant church	3.1 seconds

For the testing in this section, a group of musicians performed an acoustic arrangement of the pop song "Sunday Morning" with the following instrumentation: voice, acoustic guitar, upright bass, and a cajón. The two different miking and signal routing approaches mentioned in Section 3.1.1 were both tested and compared.

First, the testing was done through the approach using individual miking techniques for each instrument. The vocalist and all other instruments each had a microphone; no pickups or line outputs were used. The performances were accurately recognized by all three algorithms in all of the tested spaces. The test then proceeded with the other approach using one omnidirectional microphone to capture all of the music. The musicians surrounded the microphone, with the vocalist standing closer to the microphone and the percussionist a few feet further away than the others to balance the levels. The results of this approach were similarly accurate in all spaces and without any notable differences compared to the earlier approach. However, there was one exception; in the reverberant church, the tempo detection was intermittently successful. The detected tempo was sometimes unstable and shifted up or down. However, most of the times the algorithm was correct in its detection and thus was concluded to be accurate. The results of this test show that the initial signal does not have to be dry for the recognition to function properly. However, as mentioned earlier, if the signal was reverberant, it would interfere with the electro-acoustic room simulation.

3.2 Functional Consistency

After the testing of the recognition accuracy was completed, the system's consistency in its operation was tested. A performance was repeated several times in

order to observe whether the system reacted consistently. This was done with the same group of musicians and instrumentation as that in Section 3.1.5, though there were additional genres of music performed. Pop, bossa nova, and jazz were performed ten times each, and the results of the system's behavior were observed and recorded. This test was done in the room with the multi-channel speaker setup (see Figure 3.1), in order to fully test the system's functionality from music recognition to electro-acoustic room simulation.

Figure 3.1: Room with multi-channel speaker setup.

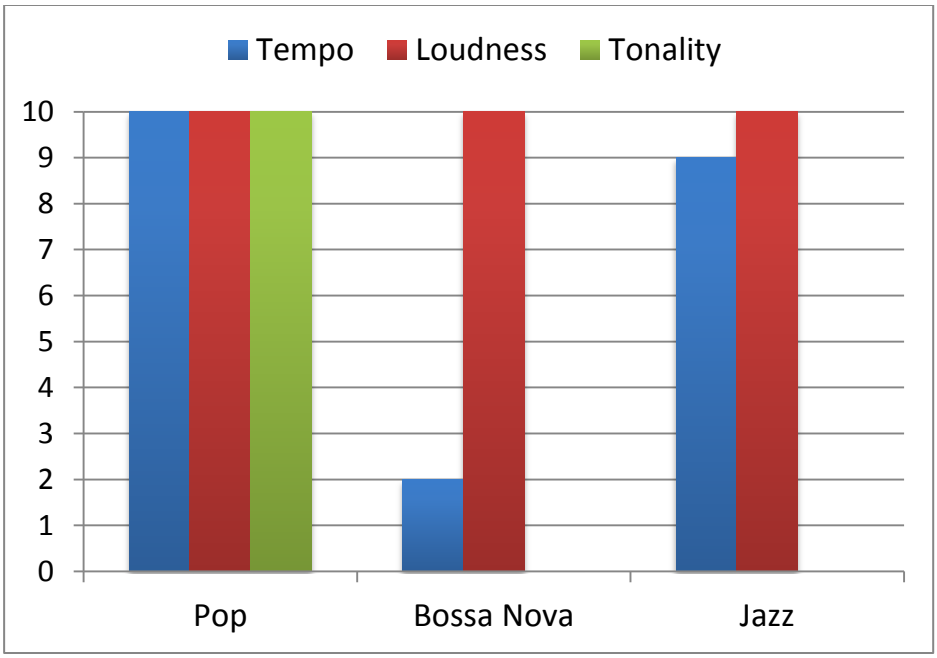

Figure 3.2: The number of accurate algorithm detections for each genre of music.

The consistency of the system is presented in Figure 3.2. The system's behavior for pop music remained constant in all of the ten rounds of performances. There were no significant deviations in music detection or in the acoustical parameter conversion process. The tonality algorithm was inaccurate in its detection for all of the bossa nova and jazz performances. Bossa nova, as indicated in the figure, had the most inconsistencies. The phenomenon that caused the inconsistency was the detection algorithm's accurate recognition of the tempo for two of the performances. The algorithm was unable to detect the tempo correctly for the other eight performances, but for an unknown reason it accurately identified the tempo in the two performances, despite the fact that no notable difference in all ten performances was observed. In the ten jazz performances, the system operated in a consistent manner except in one of the performances, the tempo algorithm counted in a tempo twice faster than the others. In the other nine performances, the tempo algorithm counted in a two-feel (every other beat of the initial tempo), but in one of the performances the algorithm counted every beat, which technically was the actual correct tempo. Other than this, inaccuracies stayed consistently inaccurate, and accuracies were consistently accurate in all ten performances.

3.3 Stereo vs. Multi-channel Output

In this section, the effect of the room simulation was compared between a stereo and a multi-channel output system. The stereo system was tested binaurally through headphones. A separate Max/MSP patch was created for testing the stereo output. The patch used a two-channel version of Peters' late reverberation (Braasch, Matthews, & Peters, 2010), and the original dry signal of the performance was also added to the stereo output mix.

According to the author's introspective review, the two output settings had a drastically different effect. The author perceived the multi-channel output system to portray a better sense of spaciousness and realism in room simulation. The binaural setting had an inevitable sound of music through headphones. This may possibly be due to the absence of HRTFs in the headphone experiment, whereas the HRTFs were available in the multi-channel experiment. Through the headphones, instead of a room-like sensation of acoustics, the artificial reverberation was much more noticeable, most likely due to it being critically audible closer to the ears. The mix of the direct signal also had a critical effect on the perception. Whereas with the multi-channel output, the dry direct sound was independently distinguishable, with the binaural output, the perceived sound was a signal with reverberation mixed to it, with no separation in the localization and discrimination of the direct and the reverberant sound.

4. CONCLUSIONS

4.1 Suitable Parameters and Settings

A single ideal setting for the algorithm conversion (from musical feature to acoustic parameter) does not exist to suit every person in every environment. Because every person may have a different aesthetic preference, the testing of this system weighed heavily on the objective factors of the accuracy and functionality of the algorithms. The author adjusted the settings to those mentioned in Section 2 according to the author's training, to the extent that this knowledge was applicable. The settings are also at a good starting position to make further adjustments if necessary. These settings are summarized in Table 4.1.

Table 4.1: Conversion settings of the system used in this thesis.

Musical Feature	Acoustical Parameter	Conversion
Tempo	Reverberation time	Linear scale of 60 – 240 bpm converted to a range of 3 – 0.5 seconds
Loudness	Direct to Reverberant Ratio	Linear scale of -96 – 30 dB converted to a range of 127 – 50 dry/wet value
Tonality	Low Pass Filter	The brightest key has a cutoff frequency of 20 kHz, the darkest key has a cutoff frequency of 10 kHz

4.2 Inaccuracies and Drawbacks

As explained in Section 3 위, the music recognition algorithms do not always detect the musical features accurately. A number of diverse conditions have been outlined in this paper, and must be taken into consideration in working with this system. The content of music, which is substantially affected by its genre, is the most important

factor in the accurate operation of the system. If detectable musical information is clearly sufficient in the music, the algorithms function more accurately. However, if the musical content becomes too complex, the algorithm becomes weaker in discernment.

Vocal music, even with rhythmic accompaniment, tends to distract the tempo algorithm because of its smooth onsets in performance and the fact that the vocal is generally mixed to be louder than the other instruments, dominating a substantial amount of the audio signal.

The "chorddetector~" function extracts chords from the music. However, the theory of the harmonic brightness (mentioned in Section 2.5.3) is technically affected by the key of the song, which is determined by the combination of the sequence of chords. Therefore, although an argument can be made that a chord is a "temporary" key, conventional music theory does not define a key on every chord of the music.

Also, as previously noted in Section 2.5.1, the tempo is not the only factor that affects a suitable reverberation time. Although slower tempos with higher reverberation times and vice versa may be perceptually pleasing, it may not be appropriate for the aesthetics of the specific genre. Therefore, the theory of converting tempo to reverberation time applied in this paper may not be suitable for certain genres.

4.3 Future Work

This thesis has ample room for growth and improvement. Numerous functions may be added and some drawbacks concerning algorithm accuracies may be fixed as well. For example, more music recognition algorithms may be incorporated in the system in order to enhance the capabilities of this system. An algorithm with genre recognition would be a feature that would facilitate the realism and application of this system.

Also, a more statistical approach may be taken for the testing of the system. This includes setting numerical thresholds in evaluating the algorithms' accuracies, in order to extensively compare the deviation of the algorithm's operation for each performance. Conducting the tests psychophysically with other test subjects may enhance the results for various perceptions as well.

Investigations on the gain levels of the system and in each of the algorithms would also be a work to be considered. Levels in each part of the signal flow have a significant effect on the other parts of the system. For instance, the gain level in the microphone preamp or within the audio input (the “demosound” object) affects the level sent to the loudness algorithm, which as a result affects the direct to reverberant ratio. Therefore, with the same loudness level of the actual performance, the level in which the loudness algorithm detects may be entirely different. The tempo and chord detection algorithms may also react differently depending on the input level. Thus, studying the appropriate gain settings may possibly lead to solving accuracy problems as well.

Another aspect to examine is the different microphone placements and/or different source locations. Although in this thesis, an omnidirectional microphone had been placed in the center of the musicians for the one of the approaches, there may be an ideal microphone placement or even musician placement.

Also, introducing the concept of panning to this system may contribute to improving the reverberation as well. Increasing the number of inputs to the system (multi-track input) may also be effective. The ViMiC system may also be helpful for this process.

This paper covers a foundation of a system, with tests in determining its strengths and weaknesses. The basic elements of the system have been researched and developed, with the purpose of altering the room acoustics based on the music performance. Further progress of this system may develop a useful tool for music performance spaces.

BIBLIOGRAPHY

Beethoven, L. (**2005**). Symphony No.5 in C minor. *The Best Classics... Ever!* EMI.

Bello, J. P., Daudet, L., Abdallah, S., Duxbury, C., Davies, M., and Sandler, M. B. (**2005**). "A tutorial on onset detection in music signals," IEEE Speech Audio Process. **13**, 1035-1047.

Bello, J. P., Duxbury, C., Davies, M., and Sandler, M. (**2004**). "On the use of phase and energy for musical onset detection in the complex domain," IEEE Signal Process. Lett. **11**, 553-556.

Blesser, B., and Salter, L.-R. (**2007**). *Spaces Speak, Are You Listening?* (MIT Press, Cambridge, MA), pp. 97-203.

Bon Jovi, J., Sambora, R., and Martin, M. (**2000**). It's My Life. [Bon Jovi] *Crush*. Island Records.

Braasch, J., Matthews, T., and Peters, N. (**2010**). ViMiC - Virtual Microphone Control for Jamoma and Max/MSP: User Manual. URL https://github.com/Nilson/ViMiC-and-friends/blob/master/ViMiC_manual.pdf. Date Last Accesed 06/13/2013.

Chappell, J. (**2013**). The Circle of Fifths Explained. URL http://www.harmonycentral.com/t5/Lessons-Theory/The-Circle-of-Fifths-Explained/ba-p/34654871. Date Last Accesed 06/13/2013.

Charpentier, F. J. (**1986**). "Pitch detection using the short-term phase spectrum," IEEE Int. Conf. on Acoust., Speech, and Signal Proc. (ICASSP 11), Apr. 7-11, 1986, Tokyo. **11**, 113-116.

Davies, M. E., and Plumbey, M. D. (**2007**). "Context-dependent beat tracking of musical audio," IEEE Audio, Speech, Language Process. **15**, 1009-1020.

Ellis, D. P. (**2007**). "Beat tracking by dynamic programming," J. New Music Res. **36**, 51-60.

Fastl, H., and Zwicker, E. (**2007**). *Psychoacoustics: Facts and Models* (Springer, Berlin), pp. 211-228.

Goldsmith, A. N. (**1944**). *Patent No. 2,354,176.* United States of America.

Green, D. M., and Patterson, J. (**1969**). "Pitch associated with an irregularity in the phase spectrum," J. Acoust. Soc. Am. **46**, 88.

Griesinger, D. (**1991**). "Improving room acoustics through time-variant synthetic reverberation," Proc. 90th Aud. Eng. Soc. Conv., Feb. 19-21, 1991, Paris. Preprint 3014.

Hartmann, W. M. (**1983**). "Localization of sound in rooms," J. Acoust. Soc. Am. **74**, 1380-1391.

ISO 226:2003. (**2003**). *Acoustics -- Normal Equal-loudness-level Contours.* Geneva, Switzerland: International Organization for Standardization.

Jackson, M. (**1982**). Billie Jean. *Thriller*. Epic Records.

Jehan, T. (**2005**). "Creating music by listening," Ph.D. thesis, Massachusetts Institute of Technology.

Jehan, T. (**n.d.**). Max/MSP software module "loudness~". URL http://cnmat.berkeley.edu/files/maxdl/OSX-MachO/loudness~_1.3.2.tar.gz. Date Last Accessed 05/03/2013.

Jot, J.-M. (**1992**). "Etude et Réalisation d'un Spatialisateur de Sons par Modèles Physiques et Perceptifs" ("Design and implementation of a sound spatializer based on pysical and perceptual models"), Ph.D. thesis, French Telecom.

Kosma, J. (**2003**). Autumn Leaves. [Eddie Higgins Trio] *Bewitched*. T. J. Comm.

Lee, D., and Cabrera, D. (**2010**). "Effect of listening level and background noise on the subjective decay rate of room impulse responses: Using time-varying loudness to model reverberance," Appl. Acoust. **71**, 801-811.

MacCallum, J. (**n.d.**). Max/MSP software module "running-average.js". URL. http://www.cnmat.berkeley.edu/files/maxdl/ALL/CNMAT-Everything-OSX-MachO.tar.gz. Date Last Accessed 05/03/2013.

Mozart, W. A. (**2005**). Die Zauberflöte (The Magic Flute), opera, K. 620: Act 2. No. 14. Der Hölle Rache. *Discover Music of the Classical Era*. Naxos.

Mozart, W. A. (**1992**). Le nozze di Figaro (The Marriage of Figaro), opera, K. 492. *Anechoic Orchestral Music Recording*. Denon Records.

Olofsson, F. (**n.d.**). Max/MSP software module "f0.average_tempo". URL http://www.fredrikolofsson.com/software/f0.abs110216.zip. Date Last Accessed 05/03/2013.

Peters, N., Matthews, T., Braasch, J., and McAdams, S. (**n.d.**). Virtual Microphone Control and other related Jamoma modules, software repository. URL https://github.com/Nilson/ViMiC-and-friends. Date Last Accessed 05/23/2013.

Rettinger, M. (**1957**). "Reverberation chambers for broadcasting and recording studios," J. Audio Eng. Soc. **5**, 18-22.

Sangster, F. L., and Teer, K. (**1969**). "Bucket-brigade electronics: New possibilities for delay, time-axis conversion, and scanning," IEEE J. Solid-State Circuits. **4**, 131-136.

Stark, A. (**n.d.**). Max/MSP software module "btrack~". URL http://www.eecs.qmul.ac.uk/~adams/software/btrack~.zip. Date Last Accessed 05/03/2013.

Stark, A. (**n.d.**). Max/MSP software module "chorddetect~". URL http://www.eecs.qmul.ac.uk/~adams/software/chorddetect~.zip. Date Last Accessed 05/03/2013.

Stark, A. M., Davies, M. E., and Plumbley, M. D. (**2009**). "Real-time beat-synchronous analysis of musical audio," Proc. of the 12th Int. Conf. on Digital Audio Effects, Sep. 1-4, 2009, Como. **DAFx-09**, DAFX1-6.

Stoess, H. (**n.d.**). History of Tuning and Temperament. URL http://www.terryblackburn.us/music/temperament/stoess.htm. Date Last Accessed 06/13/2013.

Smith, J., and Lee, N. (**2007**). Artificial Reverberation and Spatialization. https://ccrma.stanford.edu/realsimple/Reverb/. Date Last Accessed 07/02/2013.

Yamaha Corporation. (**2013**). Active Field Control (AFC) product webpage. URL http://www.yamahaproaudio.com/global/en/products/afc/systems.jsp. Date Last Accessed 06/16/2013.

Yost, W. A. (**2000**). *Fundamentals of Hearing: An Introduction* (Academic Press, San Diego, CA), pp. 193-196.

APPENDIX A: EQUIPMENT LIST

A.1 Digital Signal Processing

A.1.1 Computers

- Macbook Pro
  OS: Apple OSX 10.8.3
  Processor: 2.6 GHz Intel Core 2 Duo
  Memory: 4 GB 1067 MHz DDR3
- iMac
  OS: Apple OSX 10.8.3
  Processor: 3.4 GHz Intel Core i7
  Memory: 16 GB 1600 MHz DDR3
- Mac Pro
  OS: Apple OSX 10.7.5
  Processor: 3.2 GHz Quad-Core Intel Xeon
  Memory: 6 GB 1066 MHz DDR3

A.1.2 Software

- Cycling ’74 Max/MSP 6.0.4
- MathWorks Matlab R2012a (7.14.0)
- AFMG EASERA 1.1.0.30

A.1.3 Virtual Studio Technology (VST)

- Native Instruments Kontakt 5 (VSL library)
- Audiobro LA Scoring Strings
- Native Instruments Session Strings Pro

A.2. Signal Input/Routing Hardware

A.2.1 Microphones

- Shure KSM44
- Shure SM57
- AKG 414
- Neumann KMS105
- Sennheiser MD421
- Earthworks TC30

A.2.2 Microphone Pre-amplifier and Mixer

- ART TubeOpto 8
- Yamaha MW12CX

A.2.3 Interfaces

- RME ADI-648
- Digidesign Mbox Pro 2

A.3 Playback Devices

- Boston CS23 II with Dayton MA1240a power amplifier
- Shure SRH940 headphones

A.4 Instruments

- Martin XC1T Ellipse
- Gibson Les Paul Studio
- Ibanez GSR200
- Bellafina Model 50
- CB Percussion Drum Set with Zildjan A Custom Cymbals
- Meinl Headliner Series Cajon
- Yamaha CP 300

APPENDIX B: MUSIC USED FOR TESTING

Note: All entries are listed in the bibliography.

B.1 Performed Pieces (Live)

Title	Artist / Composer
Sunday Morning	Maroon 5
Rio Funk	Lee Ritenour
Piano Sonata No. 11	Wolfgang Amadeus Mozart
Don’t Stop Believin’	Journey
Autumn Leaves	Joseph Kosma
Silent Night	Franz Xaver Gruber
The Girl From Ipanema	Antônio Carlos Jobim
Lascia ch'io pianga	George Frideric Handel

B.2 Recordings

Title	Artist / Composer
Overture of the Marriage of Figaro	Wolfgang Amadeus Mozart
Autumn Leaves	Eddie Higgins Trio
Billie Jean	Michael Jackson
It’s My Life	Bon Jovi
Symphony No.5, Movt.1	Ludwig van Beethoven
Magic Flute	Wolfgang Amadeus Mozart