Florida International University FIU Digital Commons

**FIU Electronic Theses and Dissertations** 

University Graduate School

11-1-2019

# Augkit: an Augmented Drum Set System Designed for Live Performance

Mario A. Carvajal Florida International University, mcarva01@fiu.edu

Follow this and additional works at: https://digitalcommons.fiu.edu/etd

Part of the Music Commons, and the Software Engineering Commons

#### **Recommended Citation**

Carvajal, Mario A., "Augkit: an Augmented Drum Set System Designed for Live Performance" (2019). *FIU Electronic Theses and Dissertations*. 4343. https://digitalcommons.fiu.edu/etd/4343

This work is brought to you for free and open access by the University Graduate School at FIU Digital Commons. It has been accepted for inclusion in FIU Electronic Theses and Dissertations by an authorized administrator of FIU Digital Commons. For more information, please contact dcc@fiu.edu.

FLORIDA INTERNATIONAL UNIVERSITY

Miami, Florida

# AUGKIT: AN AUGMENTED DRUM SET SYSTEM DESIGNED FOR LIVE PERFORMANCE

A thesis submitted in partial fulfillment of the

requirements for the degree of

MASTER OF MUSIC

by

Mario Andres Carvajal Castro

To: Dean Michael R. Heithaus College of Arts, Sciences and Education

This thesis, written by Mario Andres Carvajal, and entitled Augkit: An Augmented Drum Set System Designed for Live Performance, having been approved in respect to style and intellectual content, is referred to you for judgment.

We have read this dissertation and recommend that it be approved.

Joel Galand

Orlando Garcia

Jacob Sudol, Major Professor

Date of Defense: November 1 2019

The thesis of Mario Andres Carvajal is approved.

Dean Michael R. Heithaus College of Arts, Sciences and Education

Andrés G. Gil Vice President for Research and Economic Development and Dean of the University Graduate School

Florida International University, 2019

#### ABSTRACT OF THE THESIS

# AUGKIT: AN AUGMENTED DRUM SET SYSTEM DESIGNED FOR LIVE PERFORMANCE

by

Mario Andres Carvajal

Florida International University, 2019

Miami, Florida

Professor Jacob Sudol, Major Professor

Augkit is an augmented drum set system designed for live performance. Augkit consists of a drum kit, microphone, audio interface, MIDI pad controller with foot switches, set of speakers, and a computer. I designed the processing audio engine in the visual programming language Max. Max processes the audio signal in a variety of ways that include delays, flanger, reverb, and synthesis that depend on the frequency content of the signal. Foot switches and MIDI pads toggle and modify the parameters of Augkit. This approach to playing drum set proves to be very flexible and offers new aesthetic possibilities for live performance and interaction with other musicians.

iii

# TABLE OF CONTENTS

CHAPTER	PAG
I. INTRODUCTION. A. Historical Background. B. Current State of the Industry. i. Notable Users. ii. Hardware Products.	1 1 4 4 6
II. OBJECTIVES. A. Design Goals. i. Modularity. ii. Synchronization and Flexibility. B. Aesthetic Goals. i. Balance. ii. Pitch Layer. iii. Interactivity.	9 9 11 12 12 12 14 15
III. HARDWARE	16
IV. SOFTWARE. A. MIDI Messages. B. Delay Module. C. Synthesis Module. D. Flanger and Reverb Modules.	18 18 20 23 28
V. PERFORMANCE	32
VI. CONCLUSION	35
BIBLIOGRAPHY	37
APPENDIX	39

FIGURE	PAGE
1. Electronic Drum Pad	2
2. Electronic Drum Module	3
3. Acoustic Drum Trigger	3
4. Examples of Foot Switch and Foot Pedal	3
5. Yamaha EAD 10	7
6. Sensory Percussion's Sensor	.7
7. Roland TM1	8
8. AugKit's Virtual Mixer	.12
9. Layers of Sound Diagram	13
10. AugKit's Signal Flow Diagram	17
11. MIDI Numbers for each Pad and Foot Switch	20
12. Delay Module's Graphical User Interface (GUI)	.21
13. "fmmodulator" Object	.23
14. "highsynth" Object	24
15. Distribution of Scale Tones	.25
16. "lowsynth" Object	.26
17. Synth Module GUI	27
18. Flanger GUI	.29
19. Gigaverb GUI	.31
20. Delay Module	39
21. "my_delay" Object	40

## LIST OF FIGURES

# LIST OF FIGURES (continued)

FIGURE	PAGE
22. Synth Module	41
23. "PitchDetect" Object	42

#### I. INTRODUCTION

#### A. Historical Background

The nature of the acoustic drum set has always been that of a contraption, a set of percussive instruments that used to be played by separate people but, when played by one person, form a cohesive whole. This evolution eventually led to the modern drum set which consists of a number of drums, cymbals, and auxiliary percussion such as cowbells, tambourines, and woodblocks.

In the last 40 years, many drummers have been augmenting drum sets with electronics with the goal of either replacing the sound of acoustic drums or blending them with other sounds. In both cases the sound or the vibrations of the acoustic drums are used as the mechanism for triggering samples. For example, when drummer Brendan Buckley plays with Shakira the sounds of his acoustic drums are replaced by the sounds of the drums used for the original recording of each song. During many of the songs what the audience hears coming out of the speakers are the samples, not the acoustic drums. He achieves this by triggering the appropriate sounds from a laptop computer connected to his setup. This kind of setup is the norm for drummers such as Van Romaine (Enrique Iglesias), Ilan Rubin (Nine Inch Nails), who perform in high level popular music situations and whose job is to promote a commercial recording.<sup>1</sup>

<sup>&</sup>lt;sup>1</sup> Ilya Stemkovsky. "Shakira's Brendan Buckley." *Modern Drummer*(New Jersey), March, 2019.

These sorts of electronic augmentation can be achieved by supplementing a drum set with electronic drum pads, an electronic drum module, transducers mounted on acoustic drums, as well as pedals or foot switches. Electronic drum pads (see **Figure 1**) are usually a surface the size of a small dish that are fitted with a transducer and connected through cable to an electronic drum module. The striking surface is designed to mimic the bounce of drum sticks on acoustic drums and is made of either rubber or mesh. The transducer provides the drum module information about when and how hard the pad is struck. That information can be used to trigger sounds or toggle effects generated by a computer or stored on an electronic drum module.<sup>2</sup>



Figure 1. Electronic Drum Pad.

An electronic drum module (see **Figure 2**) is a small digital box that stores audio samples that one can trigger with drum pads. Drum modules usually contain hundreds of samples of synthetic sounds, percussion instruments, acoustic and electronic drums, all pristinely recorded in a high end recording studio.

<sup>&</sup>lt;sup>2</sup> Roland SPD-SX Owner's Manual. Roland Corporation, 2011. P21.



Figure 2. Electronic Drum Module.

An acoustic drum trigger (see **Figure 3**) is a transducer that clamps or attaches to an acoustic drum, picks up the vibration from the drum head, and sends that information to an electronic drum module. Acoustic drum triggers effectively transform an acoustic drum or cymbal into an electronic drum pad, with the difference being that electronic pads are very quiet.



Figure 3. Acoustic Drum Trigger.

A foot switch or foot pedal, (see **Figure 4**) is designed to control electronic devices using the drummer's feet to toggle effects or trigger samples. Switches have only two states: on and off. Pedals such as the one pictured below behave in a similar way to drum triggers, they send information about when and how hard they are struck.



Figure 4. Examples of Foot Switch and Foot Pedal.

The hardware mentioned above is routinely used to supplement acoustic drums in modern musical situations. What is still rare is for drummers to use computers or pedals to manipulate the acoustic sound of their instruments in the way that most electric guitar players have been doing since the 1960's. Now that computers are fast and portable enough to use them for this purpose I designed a system of effects for the acoustic drums called AugKit that incorporates pedals, pads, and triggers such as those mentioned directly above so that the drummer can adjust the effect settings during performance. This new hybrid electro/acoustic system gives the drummer another layer of control of the expressiveness of the instrument. In addition, AugKit is also able to process the acoustic drum sounds to create more interesting hybrid acoustic electronic sounds.

#### B. Current State of the Industry

#### i. Notable Users

Many contemporary music improvisers are using electro/acoustic setups that process their acoustic instruments. For example, Anne La Berge<sup>3</sup> improvises with flute and electronics. Bill Laurance performs solo with an acoustic piano that is amplified and processed in real time with electronics. Wadada Leo Smith has been improvising live with trumpet and electronics since the late 1990's. George Lewis improvises with a computer system that as he

<sup>&</sup>lt;sup>3</sup>Jeffrey Glen Kaiser. "Improvising Technology : Configuring Identities and Interfaces in Contemporary Electro-Acoustic Music." PhD diss.,University of California, San Diego, 2013. 133, ProQuest Dissertations & Theses Global.

describes: "analyzes an improvisor's performance in real time, generating both complex responses to the musician's playing and independent behavior arising from the program's own internal processes."<sup>4</sup> The notion of using electronics for improvisation with acoustic instruments is not new.

Furthermore, live processing of acoustic drums is also common in concerts where the audio engineer processes the drums with different effects and filters, usually with the aesthetic goal of closely replicating the sound of a commercial recording in a live setting, or to simply optimize the sound of the drums to the acoustic environment.

One of the leaders in the area of drum set and electronics is the New York-based electronic musician Zach Danziger. Danziger uses his drums to trigger melodies, chords, samples, video clips, and to control aspects of his musical interactions with his electric bass player bandmate. In addition, in his drum and bass duo Edit Bunker, Zach controls the gating of the bass guitar sound from the drums, to give the effect of melodic lines played in unison by bass and drums.<sup>5</sup>

AugKit is inspired in part by the work of Danziger and many other related musicians with the goal to function in a more traditional popular music context

<sup>&</sup>lt;sup>4</sup> George Lewis. 2000. "Too Many Notes: Computers, Complexity and Culture in 'Voyager." Leonardo Music Journal, 33.

<sup>&</sup>lt;sup>5</sup> Ken Micallef. "Edit Bunker's Zach Danziger." *Modern Drummer*(New Jersey), March, 2018.

with some elements of "experimental music"<sup>6</sup>, such as rich and slightly unpredictable textures created by frequency modulation.

Augkit also features a virtual mixer where the drummer can toggle delay, flanger, reverb, and synthesis modules. All of these modules have user controllable levels. For example, the synthesis module creates unique synthetic tones based on whether the drummer strikes the snare or bass drum. Additional module parameters such as the waveform shape, delay subdivision, and frequency modulation are all easily controllable via foot switches or Musical Instrument Digital Interface (MIDI) pads. The technical details of these modules are discussed in greater depth in chapters III and IV.

#### ii. Hardware Products

In recent years a number of companies and artists have sought to give more expression control to drummers by creating products that process the sound of acoustic drums with effects or, alternatively, use data taken by analyzing note length and amplitude to trigger drum samples. For example, the Yamaha EAD10 (see **Figure 5**) is a system that uses a stereo microphone and a bass drum trigger sensor to add effects to acoustic drums and to trigger samples.<sup>7</sup> Yamaha includes hundreds of its proprietary drum samples in the device's memory. The unit also offers many built in effects such as reverb, equalization, wah-wah, compression, phaser, delay, flanger, and distortion.

<sup>&</sup>lt;sup>6</sup> John Cage. "Experimental music." *Silence: Lectures and Writings* 7 (1961): 12.

<sup>&</sup>lt;sup>7</sup> Yamaha EAD10 Owner's Manual. Yamaha Corporation. 2017. P10.



Figure 5. Yamaha EAD10.

Sensory Percussion by Sunhouse offers a system of sensors, pickup elements, and custom software (see **Figure 6**) that detects where and how the performer hits each drum. These sensors allow the performer to map their playing surface to control sounds, filters, effects, and pitch of the samples.<sup>8</sup> Sunhouse's system<sup>9</sup> is the one of the most sophisticated of the systems that I found during my research, but it is also the most expensive.



Figure 6. Sensory Percussion's Sensor.

<sup>&</sup>lt;sup>8</sup> <u>https://sunhou.se/sensorypercussion</u>

<sup>&</sup>lt;sup>9</sup> https://patents.google.com/patent/US20160093278A1/en

The Roland TM1 and TM2 are trigger modules marketed as an easy way to add samples to an acoustic drum set. The TM1 (see Figure 7) module is housed on a floor mountable chassis that is designed to be controlled by its built in foot switches. Sample editing such as duration, pitch and added effects is available through separate software on mobile devices and computers.



Figure 7. Roland TM1.

Despite these examples, it is still very rare for the drummer himself to be in control of the effects applied to acoustic drums. Furthermore, most drummers only add sampled sounds or blend samples with the sound of their acoustic instrument. I think the nature of acoustic drums provides rich unexplored potential for ways of digitally manipulating its different timbres and dynamic range, as well as a creative fertile playground for previously unimaginable sound explorations. Lastly, the products mentioned above offer very limited customization and are very expensive, whereas AugKit is quite afforable and easily manipulated and expanded.

#### **II. OBJECTIVES**

#### A. Design Goals

The purpose of this project is to implement an electro/acoustic drum set system designed for improvisation that can be controlled by the drummer while performing. Since drummers use both hands and feet during performance, I made most performance parameters adjustable by striking drum pads on a MIDI controller and by stepping on pedals. This approach allows for quick modifications while performing without the need to look at a computer screen or the need to use a mouse.

Another design goal is to bring some of the production elements usually associated with professional studio recording to live performance. Effects such as reverb, flanger, and delay are often added during the mixing or postproduction stages; after recording drums in the studio, to give the finished product a more polished sound. In contrast, I wanted to have some of those elements available and easy to adjust for the drummer in a live setting.

#### i. Modularity

AugKit was developed using the music and multimedia visual programming language Max, by Cycling 74.<sup>10</sup> In Max many commonly used audio tasks have been preprogrammed and abstracted into what are called objects, what would historically be referred to as "unit generators." Collections of

<sup>&</sup>lt;sup>10</sup> <u>https://cycling74.com/</u>

similar objects are grouped in libraries. Max also features an application programming interface (API). The API allows the development of third party objects called externals. In Max each program is contained in a type of canvas called a patch. Patches contain objects that can be connected to each other by virtual patch cords that allow for the interaction of the objects.

A generally recognized good practice in software design and development is modularity. Modular programming minimizes duplication, is easier to trouble shoot, more flexible, easier to reuse, modify, and expand. Modular programming is also a feature of object oriented programming. In Max, modular programming is achieved by encapsulation. For example, encapsulation allows the placement of sections of a patch within their own subpatch, this technique makes the project easier to read, more concise, and easier to maintain. Following this design philosophy, each module in AugKit is designed to do a very specific task or process and is encapsulated in its own patch. For instance, AugKit consists of five main modules: Synth Module, Delay Module, Reverb Module, Flanger Module, and MIDI Module. The Synth Module contains subpatches that control the type of wave form of the synthesized tones, which tones are synthesized when the snare is struck, and another subpatch that controls which tones are synthesized when the bass drum is struck.

Even though almost all of the software in AugKit is custom made, I did use some preprogrammed BEAP<sup>11</sup> modules for high and low pass filters, all the reverb instances, and the flanger effect. BEAP modules are computationally

<sup>&</sup>lt;sup>11</sup> Berklee Electro Acoustic Pedagogy. These modules are included in Max version 7. BEAP modules were originally designed for the purpose of creating modular synthesizers.

efficient and they fit the aesthetic needs of AugKit. In addition, most programmers use some preprogrammed modules. There is no need to reinvent the wheel. Reusing preprogrammed modules saves time, resources, and reduces redundancy.

#### ii. Synchronization and Flexibility

One of the issues that AugKit seeks to address is synchronization. The goal is to design a system that is flexible in real time. With this flexibility it can conform to the tempo of the music as well as respond to the dynamics and timbres of the instrument, even when these musical elements might be different than the one planned for performance. For example, the delay module includes a tap tempo feature that synchronizes the delay to the tempo of the music. When the tempo of the music changes during performance, the drummer can tap the new tempo with one of his feet while performing, seamlessly adjusting the delay to the new tempo.

For flexibility to function in different acoustic environments, the audio signal that goes to various modules, including the delay and the amplified unprocessed drums, each have adjustable high pass filters. This is included because, in my experience, it is crucial to have control over low frequencies to prevent feedback.

#### **B.** Aesthetic Goals

#### i. Balance

Since AugKit includes multiple layers of sound and effects, it features a virtual mixer (see **Figure 8**) to maintain the appropriate balance of all the audio layers. The purpose of the mixer is to give the drummer the ability to adjust the multiple audio layers to different acoustic environments and to be able to create different balances that are appropriate for different compositions. The drummer can control the level and toggle each effect independently, as well as the sound of the unprocessed but amplified acoustic drums.



Figure 8. AugKit's Virtual Mixer

**Figure 8** is a screenshot of the virtual mixer. In this patch, the slider marked as "1" is the global level. That is the master level of the mixer. Slider number "2" controls the level of unprocessed but amplified drums. That level is only needed if the acoustic drums are not loud enough for the particular acoustic

space. Slider number "3" above controls how much audio is sent to the delay module. Slider number "4" above controls the overall level of the reverb. Slider number "5" above controls how much of the delay effect gets sent to the speakers. Sliders numbers "6" and "7" above control the level of high and low synth tones respectively. Slider number "8" above controls the level of the accompaniment tracks or samples. Samples are useful for testing and for situations where the drummer is performing by himself. Slider number "9" above controls the level of the accompaniment track that is sent to the delay effect.



Layers of Sound

Figure 9. Layers of Sound Diagram.

As **Figure 9** demonstrates, there are many options for layers of sound that can be simultaneously heard with Augkit. These layers are comprised of: the acoustic sound, the amplified sound, the drums with reverb, the synthetic tones, the live delays, and the drums with flanger effect. In addition to these layers, the delay module also has options for FM synthesis and carrier oscillator that interact with the amplitude of the audio signal sent to the synthesis module. The details of the FM synthesis and carrier oscillator are described in detail in chapter IV.

#### ii. Pitch Layer

Another aesthetic goal in AugKit is to use the acoustic drums to trigger synthetic tones to add a new pitched and timbral layer to the acoustic drums. The synth and delay modules were designed to provide these layers. The details of these modules are discussed in depth in chapter IV.

One of the design goals is to perform real time timbral analysis of the drums to process the signal differently depending on its frequential or timbral content. The information from this analysis is used to create synthesized tones.<sup>12</sup> This analysis proved, initially, to be very unreliable since the tools developed for this purpose rely mostly on monophonic pitched content. In contrast, percussive signals are unpitched and usually contain a broad spectrum of high frequency energy. For example, there is usually a large transient (a high amplitude, short-duration sound at the beginning of a waveform that occurs in phenomena such as musical sounds, noises or speech). Transients do not necessarily directly depend on the frequency of the tone they initiate. Even on a low instrument such as a bass drum that transient energy is present and is difficult to tease out.

<sup>&</sup>lt;sup>12</sup> C. Champion and M. H. Zareei, "Am mode: using am and fm synthesis for acoustic drum set augmentation," in *Proceedings of the international conference on new interfaces for musical expression*, Blacksburg, Virginia, USA, 2018, p. 33–34.

and analyses of amplitude, AugKit is able to differentiate the specific instrument the drummer strikes. This way the drummer can play a melody with the snare and a bass line with the bass drum even when both triggering sounds come through the same microphone. The technical details of this synthesis engine are discussed in more depth in chapter IV.

#### iii. Interactivity

All the modules send their audio signals in parallel to the main output. As mentioned above, these modules can be engaged individually or simultaneously. However, the delay and synth modules also have an option where the amplitude of the signal detected at the synth module drives the vibration height of the delay module's carrier oscillator. When this option is engaged it causes frequency modulation of the audio sent to the delay module. The goal here is to modulate the delay's distortion by taking information about how hard the drums are struck. This interaction creates a more complex level of expression for AugKit since the signals from the delay and synthesized tones are not just mixed together but the information of one is used to modify the signal of the other.

By its nature, frequency modulation tends to produce complex timbres. Since the drum set is naturally an instrument with a wide dynamic range. I wanted the drummer that uses AugKit to be able to explore sounds that go from a whisper to a roar which will then, in turn, drive the frequency modulation parameters to produce new timbres based on the drummer's touch. I think that

this technique creates a very natural extension of the inherent nature of the drum set.

#### **III. HARDWARE**

The hardware setup consists of a drum set, a dynamic microphone, an audio interface, a laptop computer, a Roland SPD-SX MIDI drum controller with foot switches and pedals, in addition to a contact microphone attached to a cymbal.

AugKit uses a Digidesign Mbox 2 Mini, a one channel audio interface. A multiple channel audio interface would be ideal to process the individual drums independently but one of my design goals was to solve the technical issue of being able to trigger different tones based on the content of a single audio signal. For example, if the audio contains low frequency information such as the sound of a bass drum, the system will behave one way. In contrast, if the audio contains higher frequency information such as a snare hit, then the system will behave in a different way. The computer receives and processes all audio from the audio interface and all MIDI data from the sampling pad.

The sampling pad can do more than send MIDI messages, but for AugKit's purposes I am only using its MIDI capabilities. The contact microphone is a piezo transducer, in other words, it is a sensor that detects vibrations. The contact microphone is attached to a cymbal and to the MIDI controller, and is used in AugKit to signal changes of waveform type.



#### Figure 10. AugKit's Signal Flow Diagram.

**Figure 10** shows the direction of signal flows in AugKit. The dynamic microphone is strategically placed in between the bass and snare drums. The microphone is directly connected to the audio interface, where the signal gets amplified and converted into digital data. After this stage the signal is sent to the computer through the audio interface where it is then processed by the software. After software processing, the signal goes back to the audio interface where it is converted back into an analog signal and then sent out the speakers. The MIDI controller is only used to change and toggle parameters in the software. I use the nine available pads plus a foot switch and a foot pedal for the MIDI controller, as well as the cymbal mounted contact microphone, to change parameters in AugKit.

#### **IV. SOFTWARE**

AugKit was implemented on a 2012 Dell Inspiron 7720 with 8 gigabytes of RAM and running Windows 10 operating system. The custom software was designed in the visual programming environment Max version 7.3.5 and runs on both its 32 and 64 bit versions. Most modern computers have more than enough processing power to be able to run AugKit without any problems or issues of latency.

The software receives the audio input as well as the MIDI data. The audio input is processed and analyzed by different modules and effects such as delays, flanger, reverb, and tone synthesizers. In addition, the signal is sent to a timbre and amplitude analyzer that feeds the various modules. The software also includes a module that synthesizes different sounds depending on the frequency content of the audio signal. For example, sounds that have more low frequency content such as the bass drum will drive the low register oscillators that trigger the notes of a bass line. Relevant parameters for each effect are controlled by the aforementioned pedals and pads on the MIDI controller.

#### A. MIDI Messages

Most control information is sent to the software via the aforementioned percussion MIDI controller. The MIDI controller is connected to the computer via USB. When struck, the controller sends MIDI messages to the computer. The computer parses the MIDI messages using the "MIDI\_Module" object and

depending on the MIDI channel, triggers appropriate behavior changes in Max. These behavior changes include increment/decrement the current delay subdivision, toggle the delay, reverb, flanger, or synthesis modules on/off, change the frequency value of the carrier oscillator, change the waveform shape for the oscillators, and change the tempo within tap tempo system. This setup enables the drummer to communicate with the software almost exclusively via foot switches, pedals, and by striking the nine available pads or the one contact microphone mounted on a cymbal.

**Table 1** presents a comprehensive list of all the MIDI messages

 generated by the system and their respective functions in the software.

MIDI #	60	61	62	63	64	65
Function	Increase	Toggle FM	Toggle	Decrease	Toggle	Toggle
	carrier by	Synthesis	Reverb	carrier by	Reverb	Delay
	200 Hz	Modulator	High synth	200 Hz	(Overall)	
MIDI #	66	67	68	69	71	73
Function	Toggle Synth	Inc Delay Subd	Dec Delay Subd	Tap Tempo	Change Wave type	Toggle Flanger

Table 1 MIDI messages and their functions in AugKit.

**Figure 11** presents the specific mapping of all the pads, pedals, and contact microphone to the MIDI messages in the system on an image of MIDI pad controller itself.



Figure 11. MIDI Numbers for each Pad and Foot Switch.

#### **B. Delay Module**

The delay effect features controls for tempo or time in milliseconds, feedback, delay subdivision, and the carrier oscillator. Delay effects are usually the most effective when they are synchronized with the tempo of the music. The delay effect in AugKit is also designed to work with some of the most common rhythmic subdivisions: quarter notes, quarter note triplets, eighth notes, eighth note triplets, sixteenth notes, quintuplets, and sextuplets. In addition, I also included an adjustable high pass filter that prevents low frequencies from causing unwanted feedback. **Figure 12** (below) is a screenshot of the main controls for the delay module.



Figure 12. Delay Module's Graphical User Interface (GUI).

For situations where the tempo of the delay needs to be adjusted in real time the drummer can use the tap tempo algorithm that synchronizes the delay time with the music's pulse by tapping a foot pedal. This algorithm takes the tempo in beats per minute and calculates the number of milliseconds of delay for a specific subdivision. For example, if the tapped tempo is 60 beats per minute. The algorithm divides 60,000 milliseconds by 60 to calculate a 1000 milliseconds delay. The algorithm can then divide the delay time by the subdivision, for example, for quarter notes the delay is divided by one and for eighth notes it is divided by two. The subdivision value can be incremented or decremented by striking pads 67 and 68 respectively on the MIDI controller.

#### **Distortion Synthesis**

In an effort to add an extra layer of timbral complexity, AugKit includes the option of engaging two levels of distortion synthesis techniques in the delay

module. Distortion synthesis techniques, which include frequency modulation, use small numbers of oscillators to create spectra with many more components than the number of oscillators. Audio synthesis by means of frequency modulation (FM), pioneered by John Chowning, is "perhaps the single greatest advancement in improving the accessibility of high-quality, computer synthesized sound."<sup>13</sup> The first level of distortion synthesis in the delay comprises modulating the delay signal by a carrier oscillator. The carrier oscillator in the delay module is initially set to 0 Hz. The base frequency of this oscillator can be incremented or decremented by 200 Hz by striking pads 60 and 63. When the oscillator is incremented it gives the delay a slight distortion whose pitch and harmonic content gets progressively higher and more complex.

The second level of distortion synthesis is the option to have the delay module interact with the synth module by taking the amplitude information and feeding it to the vibrate height of the frequency modulator on the delay module. This is a form of real-time interactive FM synthesis. When engaged, the frequency modulator uses a carrier frequency of 300 Hz. The vibration height is modulated based on the amplitude of the audio that triggers the "highsynth" object. The vibration height is set to modulate over the span of four seconds. The "frequency" object (see **Figure 13**) is engaged by hitting pad 61 as mentioned above.

<sup>&</sup>lt;sup>13</sup> Charles Dodge and Thomas A. Jerse, *Computer Music: Synthesis, Composition, and Performance* (New York: Schirmer Books, 1997), 115.



Figure 13. "fmmodulator" Object.

#### C. Synthesis Module

The synthesis module produces synthetic tones based on simple oscillators. The high pitch oscillator has selectable waveform types that include sine, triangle, rectangle, and saw tooth waveform shapes. The attack, decay, sustain, and release envelope of these tones closely follows the envelope of the sound of the acoustic drums themselves, since those parameters are captured by a simple envelope following algorithm.

The synthesis module takes the signal from a dynamic microphone and processes the signal simultaneously in two ways. The first method is that the signal triggers the higher frequency synth notes with in a patch named "highsynth" (see **Figure 14**). To do this the signal first passes through a BEAP high pass filter, set to cut off frequencies below 1000 Hz. Following this, the

audio signal goes to the "PitchDetect" object. If the signal's amplitude is above a certain threshold, the "highsynth" object then triggers a note from one of the following pitches of the c harmonic minor scale:  $C_4$ ,  $E 
i _4$ ,  $F_4$ ,  $G_4$ ,  $A 
i _4$ ,  $B 
i _4$ . The relative probability (see **Figure 15**) of each sounding pitch within the "highsynth" patch to be triggered follows a probability table roughly based on David Huron's analyses of the relative probability of pitch degrees across multiple classical works.<sup>14</sup>



Figure 14, below, shows the patch for the "highsynth" object.

Figure 14. "highsynth" Object.

<sup>&</sup>lt;sup>14</sup> David Huron, Sweet Anticipation: Music and the Psychology of Expectation(Massachusetts: MIT Press, 2006), 149.



The synthesis module also includes a toggle that can turn on a reverb effect on the high synth notes to give the illusion of longer length to these synthetic tones. In my experience, I found that without the reverb the high synth tones sounded too dry and unmusical.

Similarly to the operation within the "highsynth" algorithm, the audio signal that triggers the bass line pitches passes first through a BEAP low pass filter set to remove high frequencies above 150 Hz. Following this, the audio signal goes to the "pitchdetect" object. If the signal's amplitude is above a certain threshold, the "lowsynth" object (see **Figure 16**) then randomly triggers individual pitches of the c minor pentatonic scale:  $C_1$ ,  $E 
i _1$ ,  $F_1$ ,  $G_1$ ,  $B 
i _1$ .



Figure 16. "lowsynth" Object.

The relative level and sensitivity threshold of high and low synth tones is controlled in the Synth Module GUI (see **Figure 17**). The sensitivity threshold controls the minimum amplitude that audio signal must be greater than to trigger synthetic pitches. These thresholds are adjustable and are part of the settings of the Synth Module. However, in practice, I have found that the "highsynth" threshold set to 0.99 and the "lowsynth" threshold set to 0.3 prove reliable in most settings and, only occasionally, require slight adjustment.



Figure 17. Synth Module GUI.

The aforementioned "PitchDetect" object within the synth module is an encapsulation of a fundamental frequency and pitch estimator called "fzero~". "fzero~" is a built-in object included in Max 7. "PitchDetect" performs multiple layers of analysis on the incoming signal. In AugKit, this is how the patch detects the different amplitudes that cause the synthetic tones to be produced.

"PitchDetect" has adjustable attributes for minimum and maximum incoming frequencies, period (the minimum number of samples between reporting any offsets), size (the buffer size that specifies how many samples are analyzed at once), and threshold (a peak amplitude which must be surpassed before "fzero~" reports data). After much experimentation the attribute settings that work best for my purposes are encoded on tables 2 and 3 below. Although pitch detection system was initially largely unreliable in early experiements, through thorough trial and error I have found these values make the system work extremely reliably in almost every setting I have used AugKit in.

Min Frequency	Max Frequency	Period	Size	Threshold
150 Hz	2500 Hz	256 samples	512	0.99

Table 2. Attribute settings for "PitchDetect" high

Min Frequency	Max Frequency	Period	Size	Threshold
20 Hz	100 Hz	256 samples	512	0.3

Table 3. Attribute settings for "PitchDetect" low

#### **D. Flanger and Reverb Modules**

Lastly, the flanger and reverb modules are the simplest in AugKit. For both modules, I made use of the built-in Max library BEAP modules mentioned in section II. Within AugKit the drummer has controls to toggle the effects on and off, as well as a volume slider that allows the performer to adjust how much of the processed signal goes to the speakers.

A flanging effect creates a "swishing" or rapidly varying high frequency sound by adding a signal to an image of itself that is delayed by a short, variable amount of time. A typical use of this configuration is to apply a low frequency oscillator such as sine wave to control the duration of the the delay, so that the delay's duration slowly sweep between a delay of a few milliseconds to no delay at all.<sup>15</sup> The result of adding the original signal with the time-varying delayed signal is that the frequency spectrum gets peaks and notches at certain frequencies that change throughout the cycle of the flange.

The available parameters for the flanger in AugKit include: center frequency (from 0 Hz to 2000 Hz), width (from 0 Hz to 20 kHz), rate (from 0 Hz to 10 Hz), and regeneration (from 0 to 1). Center frequency is the base frequency that is used for filtering. Width is the frequency range that gets filtered. Rate is how fast the flanger sweeps over the frequency range. Regeneration controls the proportion of the delayed signal in the output, which determines the prominence of the effect. Through trial and error I have found that the parameters that work best with AugKit are the following: center 316 Hz, width 185 Hz, rate 0.13 Hz, regeneration 0.57 %, and the level slider of the flanger effect at 86. (See **Figure 18**)



Figure 18. Flanger Module GUI.

<sup>&</sup>lt;sup>15</sup> Charles Dodge and Thomas A. Jerse, *Computer Music: Synthesis, Composition, and Performance* (New York: Schirmer Books, 1997), 303.

Reverberation can be defined as "the multiple reflections of sound in a room causing sound to be heard after all sources have ceased."<sup>16</sup> Digital reverberation simulates acoustic reverberation by a system of very short delays designed to approximate various aspects of a resonant space. The digital Reverb that I use in AugKit is called Gigaverb. It is a BEAP implementation of a classic digital reveb algorithm. In Gigaverb the available parameters are: time, size, regeneration, damp, dry, early, tail, and spread. Time refers to how long it takes for sounds to fade. Size refers to the relative dimensions of the virtual room. Regeneration controls the proportion of the processed signal in the output. Damp refers to the amount of reflection/diffusion in the room. Dry refers to the amount of unprocessed signal in the output. The early parameter refers to the amount of early reflections, sounds that arrive closely after the direct sound. Early reflections give the listener psychoacoustic information about the size of the room and the relative position of the listener to the source. Tail refers to the length of decay of the sounds after the direct sound has stopped. Spread refers to the widening or narrowing of the stereo image of the reverb tail, 0% means mono, 100% means stereo.

Through trial and error I arrived at the following settings which create the impression that the acoustic drums sound larger than they are: time 4 seconds, size 60, regeneration 0.1, damp 0.5, dry 0, early 0.25, tail 0.25, and spread 0. (See **Figure 19**)

<sup>16</sup> ibid P 436



Figure 19. Gigaverb GUI.

#### V. PERFORMANCE

Electro/acoustic systems, i.e. systems that combine electronic sound with acoustic sounds in any manner, commonly fall into these three categories: 1) the acoustic instrument is played by one person and the electronics are controlled by another; 2) the electronics are programmed by a person and respond to the playing in a predetermined way that is not modified during performance; 3) the performer simultaneously performs and controls/modifies the system.

Because AugKit is an example of the third category, it provides the performer with control of the level of each module. The user then needs to choose the level of each effect before performance. Once the settings are in place, the user can engage any effect individually or simultaneously with other effects. For example, the delay and reverb modules can be engaged simultaneously. More complex and interesting levels of expression are possible when engaging the delay module simultaneously with the frequency modulation component of the synth module, since the information about how hard the drums are struck is used to modulate the delay effect.

In one of my experiments I engaged the delay module and it captured the sound of the other musicians, as well as the sound of the drums. They were all delayed together, something I was not intending, but the effect worked beautifully in that particular context. Although, further adjusting minimized this effect, I think that this sort of happy accident demonstrates how AugKit can also be explored

as a system of effects for the whole band, which is something I plan to explore on future iterations.

The modern drum set is arguably not much of a melodic or harmonic instrument. It has traditionally been an accompaniment instrument that is part of the rhythm section and, as such, its role is usually primarily supportive. However, when the Synth Module is engaged, a version of the drum set emerges that adds melodic and harmonic content to the music. Given that the bass drum can trigger bass tones and the snare can trigger melodic pitches, these new elements can allow the drummer to perform new musical roles.

In my experience in many popular musical situations, the bass guitar player and the drummer purposely play very similar rhythmic patterns. When the "lowsynth" tones are engaged, AugKit mimics the role of bass player that plays absolutely in sync with the bass drum. In that sense AugKit can serve the role of a bass player and drummer simultaneously. This feature is useful for soloing situations or for sections where the instrumentation is sparse. In addition, it provides new potentials for materials where the drums and a bassist perform a novel sort of duet.

The beauty with AugKit is that it allows for many more levels of expression than what is possible with an acoustic drum set. I would argue that acoustic drum sets are still relevant because they can bring excitement and rhythmic energy to many musical situations. They are extremely dynamic instruments with a wide frequency range, from the low frequencies of the bass drum to the high frequencies of the cymbals; and they offer many options in the realms of

dynamics, rhythmic subdivisions, sound choices, and phrasing. This said, drums are highly limited on their melodic and harmonic characteristics. AugKit is an effort to expand all of these sonic palettes by adding reverb, flanger, delay, synth tones, distortion, and frequency modulation effects that can be progressively engaged by themselves or simultaneously to help bring creative performative potential and expression to the traditional drum set.

#### **VI. CONCLUSION**

When musicians combine acoustic instruments with electronic processing it usually opens a world of expression that is unavailable with the unprocessed instrument alone. With its many layers of added timbres AugKit has already added a level of expression that has renewed my interest in exploring the available musical possibilities that this system provides within many different musical settings. I also plan to work on developing other modular effects, adding much more melodic flexibility that can allow the drummer to play in a specific key or tonal area, and as well as discovering many other new possibilities that I am yet to even consider.

As of now AugKit, relies on only a one audio input channel from an audio interface. Although this is beneficial in that it makes the system quick and easy to set-up, in the future I intend to explore developing a more complex system which will rely on more audio input channels. Through this expansion, other options will become available, using many audio signals to drive different effects and combining parameters from different modules in many novel ways.

For this iteration of AugKit, I also purposely focused primarily on the more technically challenging aspect of processing the acoustic sounds and blending them with triggered synthetic tones. This said, the foundational system I developed offers the possibilities to further augment the sonic materials with sampled sounds. As mentioned in the introduction, using sampled sounds is a very popular, common, and – thankfully – technically simple technique in today's

modern drumming. In addition to the other plans, this is another technique that I plan to add in further iterations of AugKit as I continue to develop and perform it in the years to come.

#### BIBLIOGRAPHY

Cage, John. 1961. Silence; Lectures and Writings. Wesleyan University Press

Champion C. and Zareei M. H., "Am mode: using am and fm synthesis for acoustic drum set augmentation," in *Proceedings of the international conference on new interfaces for musical expression*, Blacksburg, Virginia, USA, 2018, p. 33–34.

Collins, Nick, Margaret Schedel, and Scott Wilson. 2013. *Electronic Music*. Cambridge Introductions to Music. Cambridge University Press.

Cycling '74 "Home." Accessed October 15, 2019. https://cycling74.com/

Dodge, Charles, and Thomas A. Jerse. 1997. *Computer Music : Synthesis, Composition, and Performance*. Schirmer Books.

Estibeiro, Marc. "The Impact of the Digital Interface and the Score on Controlled Improvisation When Using Acoustic Instruments in an Electroacoustic Context," in Proceedings of the International Computer Music Association, Staffordshire University, Staffordshire, UK, 2017.

Huron, David Brian. 2006. Sweet Anticipation: Music and the Psychology of Expectation. Cambridge, Mass: A Bradford Book. http://search.ebscohost.com.ezproxy.fiu.edu/login.aspx?direct=true&db=nlebk&A N=156952&site=eds-live.

Kaiser, Jeffrey Glen. "Improvising Technology: Configuring Identities and Interfaces in Contemporary Electro-Acoustic Music." PhD diss., University of California, San Diego, 2013. ProQuest Dissertations & Theses Global.

Lewis, George. 2000. "Too Many Notes: Computers, Complexity and Culture in 'Voyager." Leonardo Music Journal, 33.

http://search.ebscohost.com/login.aspx?direct=true&db=edsgao&AN=edsgcl.719 54292&site=eds-live.

Micallef, Ken. "Edit Bunker's Zach Danziger." *Modern Drummer*(New Jersey), March, 2018.

Roads, Curtis, and John Strawn. 1996. The Computer Music Tutorial. MIT Press.

Roland SPD-SX Owner's Manual. Roland Corporation. Los Angeles. 2011

Roland TM1 Owner's Manual. Roland Corporation. Los Angeles. 2019

Stemkovsky, Ilya. "Shakira's Brendan Buckley." *Modern Drummer*(New Jersey), March, 2019.

Tenney, James., Larry Polansky, and James. *META Meta (+) Hodos.* Tenney. 1988. "Meta (+) Hodos."

Sunhouse. "Sensory Percussion." Accessed October 14, 2019. <u>https://sunhou.se/sensorypercussion</u>

Yamaha EAD10 Owner's Manual. Yamaha Corporation. Buena Park. 2017

## Appendix

# AugKit Modules and Objects

The following section includes screenshots of the various Max patches that I developed for AugKit and do not include in the text of thesis. I also provide additional information about the available parameters, the signal flow contained within each patch, and – when necessary – a brief description about the content of the displayed patch.

## **Delay Module**

Parameters: Delay time (ms), Feedback, Carrier Oscillator, Delay Subdivision,

Tempo, Level.



Figure 20. Delay Module.

The delay module contains the "my\_delay" object (pictured below)



Figure 21. "my\_delay" Object.

# Synthesis Module

"Synth\_Module" Object

Inputs: Signal, Threshold high, Threshold low

Outputs: signal high, signal low



Figure 22. Synth Module.

The "Synth\_Module" object contains the "highsynth", "lowsynth", and "PitchDetect" objects.

### "PitchDetect" Object

Inputs: signal, min freq, max freq, period, size, threshold

Period: the number of samples that "fzero~" uses for each analysis.

Size: refers to the vector size used for analysis.

Threshold: If the amplitude changes this much, an onset is detected.

Outputs: fundamental, peak amplitude

The "PitchDetect" object is an encapsulation based on the "fzero~" object, a fundamental frequency and pitch estimator. The "fzero~" object estimates the fundamental frequency of an incoming, monophonic audio signal. It performs multiple layers of wavelet transforms on an incoming vector, comparing the spacing between the peaks in each.



Figure 23. "PitchDetect" Object.