Copyright

by

Jordan Taylor Walsh

2021

The Thesis Committee for Jordan Taylor Walsh
Certifies that this is the approved version of the following Lecture Recital Document:

Electronic Music for Contemporary Classical Performers: Theory and Practice

COMMITTEE:

James Gabrillo, Supervisor

Thomas Burritt, Co-Supervisor

Hannah Lewis

Tony Edwards

Adam Vidiksis

Electronic Music for Contemporary Classical Performers: Theory and Practice

by

Jordan Taylor Walsh

Lecture Recital Document

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Doctor of the Musical Arts

The University of Texas at Austin

May 2021

Acknowledgements

This project would not have come to fruition without the consistent kindness, patience, and generosity of a number of individuals.

Thank you to Dr. Thomas Burritt and Dr. James Gabrillo for acting as co-supervisors for this document. I could not have done it without you both.

Thank you to my committee members Dr. Hannah Lewis, Dr. Adam Vidiksis, and Professor Tony Edwards.

My sincerest thanks to the excellent teachers and mentors I’ve had the privilege of working with throughout my education, including Dr. Thomas Burritt, Professor Tony Edwards, Dr. Paul Vaillancourt, Dr. Matthew McCabe, Dr. Russell Pinkston, Dr. Erik Forst, and Dr. Ryan Smith.

Thanks to my family for their consistent, unwavering support throughout my education. I would not have made it without you.

Most importantly, thank you to my partner Jessica Griggs. Thank you for your support as I made noise, disappeared for hours on end, and rambled endlessly about this project over the course of several years. I could not have done this without you.

Abstract

Electronic Music for Contemporary Classical Performers: Theory and Practice

Jordan Taylor Walsh, DMA
The University of Texas at Austin, 2021

Co-Supervisors: James Gabrillo, Thomas Burritt

The introduction of electronic tools and media is a defining aspect of twentieth and twenty-first century music-making. Recording technology changed the ways we write, disseminate, and listen to music, while electronic instruments and processing tools have successfully found a prominent place in popular, classical, and contemporary music. Despite this seemingly all-encompassing reach, there is a gap in the pedagogy: contemporary classical performers often lack the literacy necessary to effectively prepare and perform electronic music. This can be attributed to the pedagogical perspective of audio technology education in the modern conservatory setting; electronic music is either taught with the engineer and composer in mind or as an elective that is overlooked by the majority of performance students.

The primary function of this document is to reimagine audio technology pedagogy with the performer in mind. This is accomplished by approaching electronic music from two distinct but complementary angles: theory and practice. Our theoretical chapter provides the reader with context for the art of electronic music making by discussing a small selection of historical figures and concepts. Once a baseline is established, we go on to define several new terms and concepts designed to aid the performer. These include *audio-experiential discordance,* a term

that grapples with the sensory contradictions that often occur in electronic performance, and two spectra to classify the levels of interactivity and electronic manipulation in a given work.

The practical chapter covers the functionality of the most common electronic musical tools. Concepts already established in the field's vast literature have been recontextualized to be taught from the top down: rather than first establishing a base level of knowledge. Instead, this document presents concepts as they directly apply to real world examples. This approach is not intended to result in complete electronic fluency, but rather to develop the problem-solving skills necessary to troubleshoot and perform with electronics.

Both of these major sections are supported by a collection of commissions called the Electronic Integration Project (EIP). These seven pieces were specifically written to exemplify common processes and challenges in electronic performance and to act as examples in this document. These works appear throughout both the theory and practice chapters, providing tangible real-world context.

Table of Contents

List of Figures

List of Tables

Introduction

It would be something of an understatement to say that technology has permeated nearly all aspects of music making and musical consumption. We listen primarily through speakers or headphones; live performance is more often than not aided by some form of amplification; even our practice as performers is informed by the ticking of the metronome. Recording technology has allowed us to document musical works in ways that would have previously been impossible, and has generated a smattering of new tools for the realization of musical ideas including synthesizers, MIDI controllers, and drum triggers. While these tools are relatively familiar to the average popular musician or composer, performers of contemporary classical music have been slow to adopt music featuring electronic elements into their repertoire.

In my experience both as a performer and an electronic music enthusiast, most tools for creating electronic music are largely taught with the composer (in the broad sense of “one who writes music”) in mind, and digital audio workstations (DAWs) are taught from the perspective of the engineer/producer. Little consideration is given to the musician who intends to use a DAW to play a fixed media track to perform with, or who desires to perform a piece with complicated live processing without necessarily learning the language. Too often I speak with performer colleagues and students who feel that this music is beyond their reach simply because of the technical requirements. Further, the functionally infinite possibilities that electronic media can provide can feel intimidating on an artistic level. The hesitation of acoustic performers to go integrate electronic media is a two-pronged problem of theoretical implications and practical execution.

This document is written for the developed musical performer who desires to integrate electronics into their practice. The learning curve of these tools can be steep, and many of them assume a baseline knowledge of music even at their most basic levels. Additionally, the theoretical implications of adding electronic sounds to live contemporary classical performance are best explored after basic listening and theory skills have been developed. While I would absolutely encourage young performers to experiment with electronics as early and often as possible, this resource is intended for those who are already comfortable as contemporary classical performers and aim to integrate these electronic tools into their existing practice.

Theory

Electronic music is the only form of music-making that entails non-human performative elements. In a performance of an electronic work, an audience might encounter an interactive program that is actively listening and responding to a performer's input -- a work that only consists of playback and features no live elements at all, and anything in between. The introduction of non-human elements to a space that has historically been a showcase for human physical achievement is significant. This is especially true in contemporary classical music, where cooperation between players and moment-to-moment decision-making are highly valued by audiences.

This section will begin by reviewing the general foundations of electronic music-making practices as informed by early practitioners including Pierre Schaeffer and Edgard Varese. Using the foundations laid by these artists, we will define some of the most common terminology used in contemporary classical electronic music circles and examine the different listening practices that are used when engaging with electronic music. With our groundwork established, we will next define several new terms and concepts in order to make sense of the various roles performers play in electronic musical performance. Lastly, we will use all of these discussions to inform the development of two spectra to help classify and define music with electronic media.

Practice

The primary obstacle for any technical instruction in electronic music is the incredible rate of change in the hardware and software employed in this music's creation. Several approaches have been attempted to remedy this issue. Manuals like *The Computer Music Tutorial*[1] by Charles Roads eschew specific instruction in any particular language, program, or hardware system, opting instead for an in-depth study of the acoustical, mathematical, and physical phenomena that play into electronic music-making. This methodology is highly informative, but not terribly useful for the casual electronic musician due to its depth and abstraction. Conversely, texts like V.J. Manzo's *Max/MSP/Jitter for Music*[2] or Adrian Moore's *Sonic Art: An Introduction to Electroacoustic Music Composition*[3] are practical, step-by-step manuals for specific programming languages. These are more applicable to the performer, but encounter a

---

[1] Roads, Curtis, John Strawn, Curtis Abbot, John Gordon, and Philip Greenspun. *The Computer Music Tutorial*. Cambridge, MA: MIT Press, 2012.

[2] Manzo, V. J. *Max/MSP/Jitter for Music*. New York, NY: Oxford University Press, 2016.

[3] Moore, Adrian. *Sonic Art: an Introduction to Electroacoustic Music Composition*. New York, NY: Routledge, 2016.

similar problem; by focusing on the detailed workings of a specific coding language, these texts fail to meet the practical needs of musicians who might simply be endeavoring to perform a composer's work.

It should be noted that each of the above resources is exceptional in its own right, and this text is by no means an attempt at an exhaustive reimagining of electronic musical pedagogy. As such, Appendix I includes brief descriptions of a variety of educational sources for the musician who would like to acquire a deeper understanding of these tools.

This document leans toward the practical, real-world application style of the Manzo or Moore, but expands the scope outward and shifts the order in which concepts are taught. Rather than establishing fundamental building blocks and exploring how they can be utilized in increasingly complex scenarios, we will explore Max/MSP, SuperCollider, and several DAWS from the perspective of the performer, only covering concepts vital to the execution of a work. These concepts will be discussed in such a way that they are applicable to other music-making tools that a musician may encounter during their careers (CSound, FL Studio, etc).

The Electronic Integration Project

In addition to this document, I have commissioned and recorded a set of seven new works for solo percussionist and electronics to help illustrate technical and philosophical concepts called the Electronic Integration Project (referred to hereafter as the EIP). These works include:

- *Inquietude*, Jonathan Andrew Smith
- *My Battery is Low and it is Getting Dark*, Brian Ellis

- *Conversation*, Caleb Evans
- *Timelapse*, Elainie Lillios
- *Monologue V: Hidden Story*, José Martínez
- *Baptism of Wind and Waves*, James W. Parker
- *Particle Wave*, Kirsten Volness

These works are referenced throughout the document and are used as examples to illustrate specific concepts. Additionally, standalone explanations of how each piece functions both technically and theoretically are available in Appendix III.

On Scope and Terminology

Our stated goal of restructuring electronic music pedagogy for performers is a lofty one, and a lone author could never hope to cover all aspects of popular performance, classical recording, installation work, and every other unique electronic discipline on their own. With this in mind, this document will focus exclusively on concepts as they pertain to what we will call *contemporary classical* performance. While the phrase "contemporary classical" is a malleable one, for our purposes it will refer to a tradition of notated Western art music primarily featuring either voices or canon Western instruments. When we introduce electronics to contemporary classical spaces, it is easy for the terminology surrounding sound to be unclear. To keep things focused, we will use the term *signal* to refer to any sound that is being performed, recorded, or altered.

This text is designed for experienced performers seeking a foundational understanding of electronic music and is best employed when the musician has attained a comfortable command of their own musical practice. While an effort has been made to exhaustively define terminology that would be unfamiliar to the acoustic performer, a performer in their early stages may find some of these concepts rather difficult. Further, I strongly encourage readers to use this document as an experiential aid rather than as a linear text. The theory and practice sections can be read independently or together, and both should be read with either listening examples (in the case of theory) or your electronic tools of choice (in the case of practice) readily available.

Part 1: Theory

Terminology and Context

To understand the integration of electronic media into performance and how it might affect an artistic product, it is important to first develop a basic vocabulary of terms in order to speak clearly about the medium. This will entail a concise discussion regarding the medium's history with a focus on influential figures including Pierre Schaeffer and Edgard Varese. It should be noted that our goal here is *not* to observe the comprehensive history of electronic music, but rather to establish an understanding of several specific historical listening and compositional practices in order to inform our discussions. For the historically minded performer, there are a number of in-depth texts listed in Appendix I.

Terminology is a difficult thing to pin down in contemporary-classical electronic music, largely a result of the everchanging technical landscape of the field and the discipline's relative infancy. The result of this vagueness is an environment where a large number of terms with minute differences are used more or less interchangeably. It is not uncommon to hear amatuer practitioners use words like *electronic*, *electroacoustic*, and *acousmatic* all in reference to more or less the same musical practice. Understanding the broad differences between these terms will not only help make sense of electronic jargon, but will additionally assist in understanding, defining, and executing one's own electronic musical work. One of the best places to start in understanding electronic musical terminology is the work of Pierre Schaeffer.

Schaeffer was a polymath whose career spanned radio work, writing, composition, and engineering. His work with the *Groupe de Recherches Musicales*, or GRM, was fundamental in the establishment of *musique concrète*: works of pre-recorded media recontextualized in new ways by creatively editing recordings (usually on magnetic tape). This definition is best understood by unpacking the complicated translation of Schaeffer's use of the French word *concrète*. While it literally translates to the English *concrete,* Schaeffer's use of the word could be better translated to "experiential", "real-world", or "non-theoretical".[4] By this definition, the primary tenant of *musique concrète* was that it was music without vagueness. In a work by, say, Brahms, the listener can judge a work on its melody, harmony, key area, form, or a smattering of other theoretical properties. In *musique concrète*, Schaeffer believed that a purer listening experience had been achieved; the consumer is not burdened with arbitrary theory, but simply with experiencing qualities of sounds.

While *musique concrète* is often used as a catch-all term for early tape music, Schaeffer's vision was for a stricter tradition. As he saw it, *musique concrète* only accounted for works built upon sounds from the real world. While these sounds could be generated by any variety of sources (musical instruments, natural sources, nonmusical man-made objects), it was vital to Schaeffer that no electronically generated sounds were included in works of *musique concrète.* Further, Schaeffer insisted that for a work to fit into his new genre, the sounds had to be modified in some way. His commitment to

---

[4] Schaeffer, Pierre, and Christine North. "Translator's Notes." In *In Search of a Concrete Music*, translated by Christine North and John Dack. Berkely, CA: University of California Press, 2013, xii.

these dogmatic principles would lead several composers in his studio to disassociate with his leadership in favor of more open-minded compositional approaches.[5]

To better understand how Schaeffer thought about this new *concrète* music-making, it is helpful to define the *l'objet sonore* / *sound object*. Coined by philosopher Abraham Moles, this term refers to a sound that is heard without regard for or interest in how it was made.[6] To listen to a sound object is to appreciate it for its inherent sonic traits, ignoring any physical or cultural associations that we may have with the sound. For example: when a smoke alarm goes off, a number of innate responses and deductions are made. We quickly become aware that it is, in fact, a smoke alarm, and that its activation implies danger (fire). Further, we begin to consider why the alarm may have gone off. It could indicate a kitchen fire, a candle getting knocked over, or any number of other events. To hear the smoke alarm as a sound object, we must ignore all of these real-world implications, and forget that the sound we are hearing is a smoke alarm at all. As far as *musique concrète* is concerned, the smoke alarm would already have been altered in some way, making its dissociation from real-world contexts easier.

Years after defining *musique concrète*, Schaeffer would borrow and alter the Greek word *akousmatikoi* to describe a wider variety of fixed media works. This term refers to Pythagoras, who would often lecture his disciples from behind a veil. The outcome, as he saw it, was a student whose focus was not on the source of the words, but on the lecture itself. Effectively, akousmatikoi is *any noise divorced from its source*.

[5] Holmes, Thom. *Electronic and Experimental Music: Technology, Music, and Culture*. 6th ed. New York, NY: Taylor and Francis, 2016, 227.

[6] Moles Abraham André. *Les Musiques expérimentales: Revue D'une Tendance Importante De La Musique Contemporaine*. Paris: Editions du Cercle d'Art Contemporain, 1960, 42.

Schaeffer would morph the word into the French *acousmatique*, referring to the obfuscation of sound source by loudspeakers when employing playback. *Acousmatic music* refers to electronic works that utilize sounds whose sources cannot be seen, and primarily concerns itself with real-world recorded sounds being removed from their sources and contextualized in creative ways.[7] To play back an audio recording of an orchestra performing Rimsky-Korsakov's *Scheherazade* or a collage of water sounds would be acousmatic. To amplify a violinist who is visibly performing for an audience would not be.

Acousmatic is often used interchangeably with *musique concrète*, but the two words cover different subsets of electronic art. Things are complicated further when we introduce the much-misunderstood term *electroacoustic*. This might be the term most familiar to the contemporary classical performer. While it gets used as an umbrella term for almost any contemporary classical music with electronics, in much the same way acousmatic does, electroacoustic as a technical term primarily refers to electronically processed acoustic sounds. It can be made either via a recorded medium or in real time, and primarily finds its use in art music traditions.

By now, it would be excusable to be turned around by the terms that have been presented, and to question why they seem to contain so much overlap. This problem pervades the entirety of the electronic music tradition to the point that entire books have been dedicated to unraveling the unhelpful terminology that has sprung up around it.[8]

---

[7] Schaeffer, Pierre. *Treatise on Musical Objects: Essays across Disciplines*. Translated by Christine North and John Dack. Oakland, CA: University of California Press, 2017, 63-65.

[8] Landy, Leigh. *Understanding the Art of Sound Organization*. Cambridge, MA: MIT Press, 2007, 9.

While this document will not definitely solve the terminology problem, it is my hope as an author to offer a simplified scale on which we can judge the music that we perform.

In our technology and progress-driven world, one could be excused for not ever asking why we would integrate electronics into musical performance. As performers, though, the question must be asked. Why are we adding an additional complication to our already labor-intensive pursuits? What value is added by the addition of electronic sounds? Step one in answering this question is to identify what kinds of sounds we are introducing into our electronic media and how they might improve or degrade our performance. Electronic media provides us with functionally limitless access to every imaginable sound. This does not mean, though, that we should employ electronic media in every instance, or that every possible sound is an advisable choice in a given work. One could make the argument that electronic media should primarily be employed in situations where a desired outcome is not reasonably achievable by acoustic means. While this ethos provides a great starting place when asking ourselves why we would integrate electronics, it does not work in every scenario.

The easiest hole to poke in this philosophy is found in the phrase "reasonably achieved by acoustic means". How far can we stretch the word *reasonably*? Say, for instance, that a composer wants to write a part for a violin that they know is not performable. This would be an appropriate use of electronic violin sound because the sound they desire is not achievable by any other means. If that part is possible to perform, though, it is more advisable to simply allow a human to play the part. Of course, this ethos raises red flags concerning accessibility, economics, and equity. What about situations where there are no violinists available, or where their rates are too high for the composer to afford? What about aspiring student musicians who might not know a capable violinist but are eager to put their work out in the world? To say that electronics are only advisable when they are the only capable tool is exclusionary to musicians who might not have any other options. While live performers are often expensive, there are dozens of free or cheap audio technology tools available online. In modern musical environments, it is not uncommon for an electronic instrument to be a composer's only available mode of engagement.

The other major hole in this mindset is best exemplified by the question that every college freshman studying electronic composition has asked their teachers: "What if I want it to sound like that?" While this inquiry often elicits eyerolls, it raises a valid question about aesthetics. Some sounds that might be considered "cheesy" or "dated" are actually desirable in certain contexts. For instance, early video game music was often made using primitive eight or sixteen bit sounds because of computational constraints. While these sounds were originally employed out of necessity, a community of musicians who grew up with them have formed what is now called the *chiptune* genre, which only employs sounds available on these older sound cards. They do not compose with less robust implements because they have to, but because they want to.

Experiencing Electronic Music

The skills employed to listen to, analyze, and unpack electronic music are fundamentally different from those used in other types of music for a number of reasons. Most obviously, music with electronics often features sounds that would not be possible on conventional or canonical instruments. Perhaps more importantly, though, is the history and heritage of electronic music and the influence it has over today's aesthetics.

No discussion of electronic music is complete without at least a cursory examination of composer Edgard Varese's contributions. Varese's *Poeme Electronique*[9] was a central component of Le Corbusier's Philips Pavilion, an architectural feature of the 1958 World's Fair. While it was by no means the first experiment with electronic tape (Varese himself had experimented with electronic media in his earlier work *Deserts*), it represented a watershed moment for electronic media. The Philips Pavilion was outfitted with well over three hundred speakers suspended in the building, each playing back one of 15 synchronized tape tracks. Perhaps more impressively, these speakers were strategically turned on and off to give the sounds a sense of motion.[10] Even judged on technical achievement alone, *Poeme Electronique* was one of the greatest endeavors of electronic music of its time.

The work also acts as an excellent introduction to Varese's larger ideas about music-making. While writers like the Italian Futurist Luigi Russolo had suggested that man-made noise was the music of the future, Varese posited that the distinctions between music and noise was a product of technical limitations. With the rapid development of new technologies, Varese thought, one could conceive of any sound possible, and the need to judge music on melody, harmony, and the like was dated at best. In *Poeme Electronique*, he finally achieved this goal. With no musicians required to interpret his music, he could carefully sculpt his desired sounds without limitation. He

[9] Varèse, Edgard. *Poème électronique.* New York, New York: Boosey & Hawkes, 1987.

[10] Holmes, 202-207.

defined this mode of composition in the phrase *organized sound*: works where interest was not found in linear harmonic progressions, but in “the movement of sound masses, of shifting planes”, which “...will flow as the river flows”.[11]

To some, this conception of music as something beyond the theory we impose upon it might sound idealistic at best and pedantic at worst. Linear harmonic music has clearly not fallen out of fashion, so Varese must have been wrong. While he may not have been entirely correct, his ideas about integrating all sounds into music-making have notably influenced modern music, in both popular and experimental circles. Sampling, for example, is a practice that is used in popular and experimental settings to make reference to a cultural icon that an audience might already be familiar with. Whether simply playing back a sampled sound clip as it is, or making alterations to it, the inclusion of the sample effectively places a nonmusical sound into a musical context. Paul McCartney’s *Silly Love Songs* begins with samples of factory sounds like whistles in place of drums, and Pink Floyd’s *Money* samples cash registers opening, coins dropping, and bank tills closing. For musicians who might feel uneasy about this looser definition of what constitutes music, it can be comforting to know that most of us have already developed these skills simply by consuming popular art.

The takeaway from Varese’s musings is that all sound is capable of being musical sound, and an ear for timbre is key to appreciating music of this kind. This idea has been well tread by thinkers like John Cage and is not particularly revelatory, but nevertheless bears repeating because of the interlocking histories of electronic and

[11] Varèse, Edgard. “The Liberation of Sound.” Essay. In *Audio Culture: Readings in Modern Music*, edited by Christoph Cox and Daniel Warner, 17–21. New York, NY: Bloomsbury Academic, 2017, 18.

experimental music-making. It was composers like Cage, Oliveros, and Varese who stretched the sonic capabilities of new music technologies to their logical conclusions. The experiments they performed laid the groundwork for not only the contemporary classical composers of the future, but for popular music as well.

A substantial portion of this section has been dedicated to an explanation of Schaeffer's theories on sound objects and *concrète* music. While this is appropriate due to his monumental status in the history of electronic music, it would be irresponsible not to address the issues with his dogmatic theories on composition. Indeed, Schaeffer occupies a slightly Wagnerian space in electronic music circles; monolithic, but not without problems. One of our prime examples, *Poème électronique*, does not even attempt to fit into Schaeffer's system but has become the most prevalent example of early fixed media music. While foundational, Schaeffer's incredibly strict adherence to sound objects as compositional devices has never been the norm in electronic music making, and in fact his position fails to account for the realities of how humans interact with sounds.

Schaeffer believed that the correct way to listen to fixed media music was to remove as much real world context as possible, appreciating it for only its timbral qualities, calling the approach *reduced listening*. This method of listening is not inherently problematic. There is inherent aesthetic value to all sounds, and there is no problem if a listener finds that they are best able to appreciate a work for its sonic qualities alone. The issue is more that Schaeffer insists that his way is singularly correct. While true *concrète* listening is a valuable method of engagement, it is also a difficult thing to do on a biological level.

A vast majority of human responses to stimuli can be traced to the survival tactics of early people. Just like an early human might have heard a predator and inferred danger, a modern human hears a smoke alarm and responds accordingly. Real world context in listening is a survival mechanism that is not so easy to turn off. While Schaeffer's theories on acousmatic listening are viable, they are not something that the average listener is immediately capable of doing (or even has an interest in learning to do).

This point is best expressed by psychologist and musicologist Luke Windsor in the essay *Through and Around the Acousmatic: the Interpretation of Electroacoustic Sounds*. Windsor proposes an ecological approach to assessing human perception, defining several terms as they relate to both musical and general environments. The first and most important of these is affordance. Says Windsor: "Objects and events are related to a perceiving organism by structured information, and they 'afford' certain possibilities for action relative to an organism. For example, a cup affords drinking, the ground, walking".[12] Our smoke alarm example from earlier affords us an awareness of danger, and suggests that we remove ourselves from it. Affordances can take place in real or virtual environments; a real environment being one where events are directly related to their sounds, and a virtual environment being one where the event that would produce a sound has not actually occurred. By this definition, all acousmatic environments are virtual in nature.

[12] Windsor, Luke. "Through and Around the Acousmatic: the Interpretation of Electroacoustic Sounds." Essay. In *Music, Electronic Media and Culture*, edited by Simon Emmerson, 7–35. New York, NY: Routledge, 2016.

Using Windsor's framework, we can recontextualize Schaeffer's reduced listening strategy in a more quantified manner. By insisting that listeners ignore real world connections that they might perceive when listening to acousmatic music, Schaffer is suggesting that they ignore affordances when listening in a virtual environment. While on the surface this seems like a reasonable proposition, Windsor concludes that this is not how virtual environments function. While the environment of the acousmatic work may be virtual, the environment that the audience occupies (concert hall, their home stereo system, et cetera) is real. When sounds are produced in a virtual environment, the real environment in which they are perceived has an effect on the listener's experience.

So if a virtual environment does not necessarily reduce the presence of affordances, what effect does it have? Windsor proposes that there are two parallel possibilities: literal affordances and interpreted affordances. In a literal affordance, the listener hears sounds as they (more or less) exist in our world, even if they have been altered in some way (i.e. "that is the sound of the smoke alarm"). Interpreted affordances are more open-ended: they occur when a listener cannot necessarily place a sound in a work and they make a connection to a more familiar sound (i.e. "that kind of sounds like a smoke alarm").

Moreover, Ambrose Field points out Schaeffer's theories are, while logical, somewhat redundant. In making such a dramatic attempt to distinguish sound object-based listening, he effectively "reinvented the note".[13] Schaeffer's insistence on defining

[13] Field, Ambrose. "Simulation and Reality: the New Sonic Objects." Essay. In *Music, Electronic Media and Culture*, edited by Simon Emmerson, 36–55. New York, NY: Routledge, 2016.

every aspect of how a sound behaved led him to quantify sound objects based on their amplitudes, frequency, and length, leading him to a value system that resembles traditional Western notation systems to a somewhat comical degree.

Understanding that audiences are not likely to disengage from the real world context of an electronic work is key in choosing repertoire as a performer. If an audience recognizes a sound, they are likely to have an emotional reaction to it. If they do not, they might engage by likening it to something that they do recognize. Whether or not a listener comprehends where a sound might have come from directly impacts their perception of a work.

Take, for example, Russell Wharton's *Deus Ex Metronome*[14] for solo snare drum and fixed media. The work's electronic media is entirely sourced from a Dr. Beat DB-90 metronome. This is something that musicians are intimately familiar with, but a general audience might not be. This has a drastic effect on the work's perception. A musician in the audience might immediately understand and relate to the reference that the composer has made, and their reception of the work will be accordingly influenced by their relationship with the metronome. The casual listener might not be aware of the DB-90, or even of metronomes at all. Even if they are informed by program notes or other sources, their lack of direct personal experience with the sound source will influence how they appreciate the performance.

Another apt example to consider is *Electric Counterpoint*[15] by Steve Reich, in which a performer plays along to a track featuring recordings of several other guitar parts, ideally with the same tone as the live performer. This differs from *Deus ex*

[14] Wharton, Russell. *Deus Ex Metronome*. Portland, Oregon: Tapspace Publications, 2019.
[15] Reich, Steve. *Electric Counterpoint*. New York, New York: Boosey & Hawkes, 1987.

Metronome in that its electronic elements are recordings of a real world instrument. The music could be (and sometimes is) performed by a live ensemble of guitarists, whereas Wharton's fixed media track features a nonmusical object that was heavily altered in post production. In *Electric Counterpoint*, the listener is not challenged by the prospect of hearing a sound they might not recognize. Not only does the fixed media feature a well-known musical instrument, but further context is provided by the live performer playing the same instrument.

Schaeffer's approach to listening is made more difficult still by the introduction of easily identifiable sounds like the human voice to electronic media. The ur-example of this might be considered to be Paul Lanksy's *Idle Chatter.*[16] An experiment in granular synthesis and stochastic composition, *Chatter* features recordings of human voices being granulated into unpredictable rhythmic patterns and long choir-like chords. Lansky made note in the years after its release that listeners heard different words and phrases peeking through the texture, and found their ear's attempts to make sense out of gibberish fascinating.

More recently, Mark Applebaum's work has made thematic use of the concept of affordances. *Aphasia*[17] is a work for fixed media and an actor, who is tasked with executing predetermined motions in time with a fixed media track. The work refers to an affliction where a person partially or completely loses their ability to understand or express speech. The fixed media for *Aphasia* is made up of recordings of Applebaum's own voice making a variety of gibberish sounds. This changes toward the end of the

---

[16] Lansky, Paul. *Idle Chatter*. CD. *More Than Idle Chatter*. Naxos Digital Services US Inc, 1994.

[17] Applebaum, Mark. *Aphasia*. Menlo Park, CA: Self Published, 1987.

work where Applebaum starts counting upward in several languages, stopping when the English speaker reaches one hundred. The work's thematic material is related to its namesake, a condition that renders a person unable to communicate with others. Applebaum uses the human tendency to search for meaning in sound to reinforce an aesthetic.

Stepping even further outside of the realm of traditional *musique concrète*, Applebaum's *Pre-Composition*[18] is another fixed media work that employs human voices. Rather than altering them like Lansky, Applebaum's voices are completely intelligible, and make a meta-commentary on the tropes surrounding fixed media compositional practices. *Pre-Composition* wholeheartedly embraces the human tendency to listen for context, communicating with the audience directly via English text.

Performers ultimately benefit from audiences' tendencies to listen for context. If we are performing on a canon instrument, the odds are good that the traditional elements of music theory (harmony, melody, form) apply to us in a given work. This is where a conflict between musician and machine comes into focus. While the electronic elements in a piece might be better judged with timbre as a focus, the performer is probably working with pitches and rhythms in the normal sense. This further reinforces our rejection of Shaeffer's dogmatic principles on listening. My personal response to this conflict is to meet in the middle and listen with ears open to timbral quality, although not exclusively so.

[18] Applebaum, Mark. *Pre-Composition*. CD. *Intellectual Property*. Innova, 2010.

We as performers of contemporary classical music have a tenuous relationship with audio technology. It is no doubt a driving force in our practice and the dissemination of our work, but when those tools enter our performance spaces, they bring with them a sense of anxiety. I have too often encountered makers and consumers of contemporary classical music who feel that the integration of electronic elements into performance detracts from the expression and validity of the music. Even elements as simple as light amplification are undesirable to certain connoisseurs of the artform. This anxiety ultimately reflects the value set of the consumer, and begs the question: what is important in a musical performance, and how do electronic elements influence performing and listening experiences?

In order to answer this question, we must first define what exactly constitutes a performance. In a broad sense, a performance is the execution of a set of tasks by a human. Musically, this will generally involve an instrument, and the successful manipulation of that instrument's sonic capabilities. A musical performance can involve any number of humans on any combination of instruments, with a great importance being placed on the interactions between different subsets of the group, or the more granular interaction between the performer and their instrument in the case of the solo performer. The running theme of this definition is the humanity of performance; to a great many listeners, engaging with a person accomplishing a technical feat in real time is the height of the musical experience.

With this in mind, it is quite easy to identify the "problem" with electronic elements in live performance. If we place value on the human elements of art making, the introduction of non-human elements is understandably contentious. There are two actions that a performer can take to better their understanding of the electronic music that they might engage with, and in turn communicate that understanding to an audience. The first is to acknowledge the historical context into which electronic music as we know it was born, and to understand the effects this background has on its aesthetics. The second action is to establish a vocabulary that allows us to concisely understand and categorize different types of electronic contemporary classical music from a practical standpoint.

In contemporary classical electronic music, real world sounds are often processed, altered, and recontextualized in creative ways. With Windsor's writings in mind, we already know that we cannot completely remove real-world context save for the most extreme instances of processing, and even then listeners generally attempt to connect the sound to something they are familiar with. For most listeners, it is this context game (what Landy refers to as the "something to hold on to factor"[19]) that makes a piece of acousmatic / electroacoustic music interesting. If we accept that a listener is searching for context, then we can state that the strategic allotment or denial of that context is a useful, dramatic tool.

Luckily, writers in ludology have already done the legwork. In 2007, game designer Clint Hocking coined the phrase *ludonarrative dissonance*. Writing about the game *Bioshock*[20], Hocking asserted that the game's story was in direct conflict with the way in which the player interacted with the game. A cautionary tale against Randian Objectivism, *Bioshock's* plot speaks against self-interest as a political stance while its

---

[19] Landy, Leigh. *Understanding the Art of Sound Organization*. Cambridge, MA: MIT Press, 2007, 26-35.

[20] Irrational Games. *Bioshock*. 2k Games. PC/Mac. 2007.

gameplay follows traditional first-person shooter trends, actively promoting self-interest. The result of this dissonance is that *Bioshock's* fictional and thematic elements were out of touch with what it required players to do in order to progress.

While ludonarrative dissonance is a term that refers specifically to game design, the core concepts of interaction and expectation can be applied to almost any artform. In a 2017 video essay, film and pop culture critic Dan Olson recontextualized the term as *cinemanarrative dissonance*. Olson applied the idea to how a character in a film is treated by the camera versus the text. If a character is written one way but the camera perceives them another, a dissonance similar to what Hocking described is achieved.[21]

This term can just as easily be applied to sonic artforms like music simply by substituting criteria. Instead of conflict between gameplay and plot or text and camera, in music this dissonance is best thought of as conflict between eyes and ears. If a performer plays a violin acoustically and it sounds like what we believe it should, the audience is met with a cognitively consonant experience. If we introduce electronic processing and pitch shift the violin, the expectations of the audience will have been subverted in some way and will produce a more dissonant experience. In this example, our audience's eyes have told them something different than their ears, causing what we will refer to as *audio-experiential discordance;* the word discordance has replaced dissonance to account for the latter's existing prevalence in musical terminology.

Having defined our own version of ludonarrative dissonance, we should double back and establish that it is not an inherently negative thing. Games like *Spec Ops: The*

[21] Olson, Dan. *Ludonarrative Dissonance*. *YouTube*, 2017. https://www.youtube.com/watch?v=04zaTjuV60A&t=139s.

Line[22] have used ludonarrative dissonance to great effect, guiding the player to make gameplay choices that they will later be admonished for by the narrative. The message the game sends with its systems is in direct conflict with the message it sends with its narrative. *Spec Ops* has turned ludonarrative dissonance on its head and used what is normally a pejorative term to great artistic effect. This sort of positive dissonance might be the exception in game design, but is central to electronic music. When we experience a work with a live performer and live electronics, we are taking in two conflicting messages. Our eyes are telling us what the performer should sound like based on our expectations for their instrument, but our ears are hearing sounds that can range from close to the instrument's sound to completely alien. Even in works with fixed media where our eyes do not feed us information about what we should be hearing, there is still an element of this discordance in play. Windsor's work on ecological affordances in acousmatic / electroacoustic music states that humans will attempt to find meaning in sounds that they hear. I posit that the drama of much contemporary classical electronic music is derived from audio-experiential discordance.

The Digital Dark Age: An Aside

One of the unique challenges facing electronic music is the ever-changing nature of its tools. While instruments like violins and flutes have certainly evolved in the past 200 years, their designs are more or less the same as they were in the time of Brahms

[22] Yager Development. *Spec Ops: The Line*. Epic Games. PC/Mac. 2012.

(at least to the point that music of several hundred years ago can be performed somewhat accurately today). Conversely, some tools used for making and playing back electronic music from 20 years ago are difficult to get running today, and will continue to degrade as technology changes. This problem has been crystalized by the term *digital dark age* by scholars like Ted Kuny, who posits that the digital nature of our data-archiving methods will eventually make vast swathes of that data unreadable.[23]

This problem is directly applicable to electronic music of all kinds. When the British rock band Radiohead sampled Paul Lansky's early work *mild und leise* for their album Kid A in 2000, they were forced to work with the release version of the piece because the master files were generated on an old computer (an IBM 360/91 at Princeton) and were inaccessible. The problem haunts music with live electronic processing even more; composers have to update code in programs like Max every time the platform gets updated, and some of the earliest code written for live electronic processing is now either defunct or difficult to get running. Famously, Pierre Boulez's *Anthemes 2* has no patch associated with it, and the score ships with a massive manual to instruct the user on how to create their own digital signal processing.[24]

It is entirely possible that the electronic music of today will lack the longevity of Beethoven and Mozart because of the decay problem, and that works will disappear to history as technology changes. This is especially true of lesser known works. While a relatively unknown composer of acoustic music might leave behind scores that can be

[23] Kuny, Ted. "A Digital Dark Ages? Challenges in the Preservation of Electronic Information." *Audiovisual and Multimedia Joint with Preservation and Conservation, Information Technology, Library Buildings and Equipment, and the PAC Core Program*. Lecture presented at the 63RD IFLA Council and General Conference, September 4, 1997.

[24] Boulez, Pierre. *Anthemes 2*. Vienna, Austria: Universal Edition, 1997.

reinterpreted and reprinted, a composer of electronic music might leave behind code that stops working after an update or a digital file that wasn't properly archived and has decayed. While this might seem like a defect of the medium to a classical musician, I personally see it as appropriate, even poetic. The shelf life of electronic works (especially those with live processing) bring them a sense of life. They will not be available forever, making their performance that much more profound.

What We Do and How We Do It

Electronic music is an umbrella term that encompasses a massive variety of musical styles, instruments, and creative tools. The performance of electronic music can employ systems as simple as playback, as reciprocal as live instrumental processing, and as complex as live algorithmic composition. If we are to perform a piece of music with any of these systems in place, it is imperative that we understand the function of the electronics, the performer, and the implications of the two interacting. To do this we will define two scales: the **Interactivity Spectrum** to quantify processes, and the **Effect Spectrum** to evaluate outcomes. To put it another way: *how* is a sound being made or altered, and *what* has happened to that sound? It is important to clarify that the function of these spectra is not to indicate the value of a work. A piece which features no interactive elements and little audible alteration is of equal intrinsic value to one that features a large degree of electronic interactivity and heavy processing. Rather, these scales are intended to define the roles that nonhuman elements play in a given work so that we as performers can better understand our role.

The Interactivity Spectrum

Degree 1: The Uninteractive

The oldest and perhaps most common form of electronic music is the predetermined and uninteractive work. This is generally a work for playback, sometimes with a performer involved and sometimes without. These works have historically been referred to as Tape Pieces, referencing the compositional practice of manipulating magnetic tape to a creative end. More recently, descriptors like Fixed Media and Track have become far more standard as a response to the advent of digital playback. For our purposes, I adopt the phrase *Fixed Media* to refer to this variety of electronic music.

Uninteractive music is where the low value of electronic music to classical performers is most common. If our job is to interface with our instrument and with each other, what possible good is a piece of music which has an entirely predetermined outcome? The answer to this admittedly valid question can take two different forms depending on whether or not a human is involved in the performance.

Remembering that a musical performance's value is often judged on the quality of interactions between performers (or a player and their instrument in the case of a solo work), uninteractive music with fixed media is explained easily. In this paradigm, a work for soloist and fixed media is an interaction between two forces: the player and the track. Further, a work for ensemble and fixed media could be said to integrate the electronics as just one more performing force among many. However, this "chamber music" definition does not stand up to scrutiny for the simple reason that the fixed media *cannot respond or adapt to the player*. It is by definition unchanging, unresponsive, and predetermined. This is not *interactive*, but *reactive* music-making. As performers, we are beholden to a predetermined musical trajectory. We have little flexibility in terms of tempo, and our dynamic must be such that we fit whatever role we play in a given piece of music. We are even more constrained by music with a precise rhythmic texture, where we must often perform with a metronome in order to fulfill our function.

Given these constraints, a contemporary classical musician's aversion to fixed media music is comprehensible. If we cannot make real-time choices, then why are we even here? This reasoning is logical but reductive, and assumes without question that the centerpiece of a musical experience is the performer. In a fixed media work, I argue that the human element and their technical capabilities are not only not the focus, but are distinctly secondary to the fixed media. This requires a sound-centric approach to listening rather than the more common performer-centric one. When a human interfaces with a sonic force than cannot respond to our actions, we are completely beholden to that force. This does not detract from the human element of music-making, but changes the types of choices the performer must make. For instance, we are still beholden to the acoustic properties of the performance space, but now we must also account for the addition of the fixed media in that space. We must consider *where* the fixed media is coming from, what our role in relationship to the track might be, and how to best embody the character of the music. These considerations are all familiar to the chamber musician, but the immovability of fixed media makes it a distinct kind of reactivity: the performer exists in a system outside of their control, and the quality of a performance can be gauged by their ability to operate within that system.

Some of the oldest pieces for electronics and performers are degree 1 works. Karlheinz Stockhausen's *Kontakte*[25], for example, is written for percussion, piano, and four channel fixed media. Stockhausen's use of electronics in *Kontakte* allowed him compositional control over nearly every sonic element at his disposal, and helped him inch closer to his stated goal of achieving total serialism.[26] Stockhausen's exertion of creative control via electronic media reinforces the somewhat unbalanced nature of uninteractive music. In a performance of *Kontakte*, the musicians are entirely at the mercy of Stockhausen's fixed media track, and by proxy Stockhausen himself. Perhaps even more famously, Mario Davidovsky's *Synchronism No. 6*[27] is a Pulitzer prize winning work for piano and fixed media that exemplifies the significance of uninteractive electronic music. Davidovsky's piece is played without the aid of a metronome, meaning that the performer must be intimately familiar with the fixed media track. The piece also heavily features the concept of audio-experiential discordance; the very first note played by the pianist is mimicked backwards by the electronics, resulting in a sound that would be impossible on piano alone.

EIP: *Conversation*

The most pertinent example of uninteractive music in the EIP is Caleb Evans' *Conversation*. The work begins purely acoustically, with the first electronic sound coinciding with the first use of the bass drum (the bass drum will continue to be used as

---

[25] Stockhausen, Karlheinz. *Kontakte*. Kürten, Germany: Stockhausen Verlag, 1960.

[26] Stockhausen, Karlheinz. 1962. "The Concept of Unity in Electronic Music (Die Einheit der musikalischen Zeit)". Translated by Elaine Barkin. *Perspectives of New Music* 1, no. 1 (Autumn): 40.

[27] Davidovsky, Mario. *Synchronism No. 6*. Philadelphia, PA: J.W. Pepper, 1970.

a sort of false “trigger” for electronic sounds and sectional transitions throughout the work). The performer primarily functions as a drummer in the literal sense, providing a backbeat to an electronic track that gradually increases in both density and intensity until reaching a climax and deteriorating in a similar manner.

Degree 2: Altered Sound

This second degree of interactivity might be the most familiar to the commercial music consumer: altered sound. Here, any number of direct changes are made to an audio signal, including (but not limited to) distortion, compression, flanging, phasing, delay, and reverberation. Processed sounds are best exemplified in the electric guitarist’s pedalboard. The music being made is still directly and physically linked to the human performer and their instrument of choice, but a chain of processing units are put in place to make carefully (or not so carefully) calculated alterations to the sound.

Altered sound is easier to justify on the human-centric listening model than uninteractive music because the actions of the performer are still centered in the audience's experience. Nigel Westlake’s *Hinchenbrook Riffs*[28] for Guitar and Digital Delay offers an excellent example of this phenomenon. Westlake’s score calls for a delay pedal to be employed for most of the performance, and much of the piece’s interest is generated by the harmonic results of the performer playing over their own delayed signal. It is made clear to the audience that while the signal is being altered via electronic means, the sounds that they are experiencing are always being directly generated by the activity of the performer.

[28] Westlake, Nigel. *Hinchenbrook Riffs*. Sydney, Australia: Rimshot Music, 2003.

While instances using guitar pedals fit perfectly into the altered sound category, they might not be the most relatable examples for the average contemporary classical performer. Let us look instead to Kirsten Volness' *Particle Wave*[29] (2020) for vibraphone and Max/MSP. Movement 1 of the work utilizes a number of prebaked effects, primarily reverb and digital delay. These effects do not alter the fundamental sound of the vibraphone that the audience hears, but rather supports and reinforces that sound. If one were asked to identify the sounds being produced in the hall, they might say "reverberant vibraphone" or "vibraphone with some kind of echo". They are identifying the vibraphone as the central source of their experience. Further, these reverb and delay effects are largely static: they behave in predictable ways and do not interact with the performer. If an audience perceives the electronics to be responding in a dynamic way to the performer, they have encountered the third degree of interactivity.

EIP: *Particle Wave*

Kirsten Volness' *Particle Wave* is an excellent example of our second degree of interactivity. The work's first movement begins acoustically, and is only lightly treated with delay and reverb at certain sections. This makes it apparent to the listener that the electronic sounds are being generated by the performer at all times. Additionally, the electronic processes are predictable and do not alter the vibraphone beyond recognition, making *Particle Wave* a perfect example of altered sound.

[29] Volness, Kirsten. *Particle Wave.* Self Published, 2020.

Degree 3: Interactive Music

Interactive electronic music is perhaps the easiest to justify within the human-centric listening model. Here, a computer is listening to a live performer/performers, applying any number of live processing techniques to the outcome. The defining factor in this degree of interactivity is the staticness of the electronic processes. Look back to our delay examples. Traditional looping using digital delay falls squarely into the Altered Sound category because its outcomes are predictable, and the actions of the performer do not influence the behavior of the process. If we were to give the performer control over aspects of the delay line (time, level, repeats) with a tool like a motion tracker, it would become truly interactive music; the performer would not only be supplying sound to an electronic process, but would also exercise control over the parameters of that process. Contrarily, interactivity can also be achieved when the performer has little to no control over electronic elements, and are beholden to a process that they cannot predict. This might include dynamically changing parameters in a patch, or even third-party information adjusting a patch's input (for example, a work where Twitter feeds are parsed for data).

It is worth stating explicitly that the tools used in Altered and Interactive Music are often the same or similar, and that it is more *how* these tools are used that defines their place on the spectrum. While music using a digital delay to create a predictable echoing effect fits into the altered category, a piece that employs several delay units with dynamically changing parameters and balances might be considered interactive. The defining factor of the interactive category, then, is its dynamism.

EIP: *My Battery is Low and it is Getting Dark*

Brain Ellis' *My Battery is Low and it is Getting Dark* (2019)[30] is an adaptable solo for keyboard percussion and Max/MSP which has no consistent score. Instead, Ellis has set up an algorithm that utilizes both predetermined and user-input variables to write a new score for each performance. While one could theoretically reuse the score for *Battery* for multiple performances, this would be in conflict with the thematic elements of the piece, which concerns the last message received from the Mars Rover Opportunity in 2019. The work's interactivity stems from both the user's input in generating each score and the lack of predictability in the electronic media during a given performance. Ellis has made use of the unreproducible nature of interactive music to reference the unpredictability and melancholy nature of the rover's long life and slow demise.

Degree 4: Algorithmic Music

The final degree of interactivity concerns self-generating or algorithmic music. In contrast to the degrees that have been presented thus far, algorithmic works tend to be concerned primarily with process. The composer of an algorithmic piece has not set up discrete events to take place in a specific order as in traditional acoustic music, but has developed a self-generating system that achieves a desired aesthetic. Because of this stark difference, algorithmic works are often accompanied by some kind of meta-commentary about their semi-autonomous nature.

One does not have to look far to find algorithmic music that serves similar thematic functions. The first movement of Laurent Durupt's massive percussion trio

[30] Ellis, Brian. *My Battery Is Low And It Is Getting Dark*. Brooklyn, NY: Self Published, 2019.

Spirales (2017)[31] is written by a repetitive algorithm that cycles every 43 years. The program takes the date and time of the performance into account, and generates the score based on its place in the 43-year timeline. Thematically, this coincides with Durupt's age when his first child was born, with the piece representing his life up to that point. It should go without saying that Durupt did not compose 43 years of music personally, but set up an algorithm to compose the piece. Further, the work's trajectory takes place on such a massive temporal scale that it would be incredibly unlikely that any audience would experience the same iteration of the work twice.

EIP: *Monologue V: Hidden Story*

Monologue V: Hidden Story[32] by Jose Martinez is a semi-algorithmic work based on the experiences of mixed-race individuals. The work is more predetermined than *Battery* or *Spirales*, with a large majority of the notes being written in a conventional linear fashion. The algorithmic portion of the piece takes place exclusively in the electronics; Martinez's patch generates new music from predetermined samples each time it is performed, which the performer is then asked to mimic on their drum. This puts the performer at the mercy of the electronic element much in the same way they are in non-interactive music, but with a degree of unpredictability generating the drama of the performance. The act of learning from the unpredictable electronics mirrors the difficulty that the children of immigrants have reconciling their cultural inheritance. Martinez used

[31] Durupt, Laurent. *Spirales*. Paris, France: Self Published, 2017.

[32] Martínez, José. *Monologue V*. Austin, TX: Self Published, 2021.

the living nature of algorithmic composition to represent the contradictory social anxieties of being a mixed-race person.

The Other Axis: Physicality

Thus far, our discussion has primarily centered on digital means of reproduction. This is resultant of digital audio's dominance in the modern field of contemporary classical performance, but there is another more analog approach to incorporating electronics into music-making that predates the advent of digital media. There is already a healthy discussion surrounding this music that focuses on the analog nature of the work, but I find that simply referring to works with tape, turntables, or feedback as analog fails to capture the uniqueness of this kind of work (especially in a practice where digital tools have largely become the norm). For our categorical purposes, I propose the umbrella term *physical electronic music*.

Physical electronic music encompasses a broad range of mediums and practices, from magnetic tape manipulation to feedback music. The unifying factor is that there is a tangible interaction between the performer or audience and the sound-generating device(s). This differs from playback in that the device is conveying an experience unrelated to itself during playback. In physical electronic work, the device is part of the performance and exudes a tangible, real-world influence over the creative product.

Consider *I of IV* by Pauline Oliveros (1966). Here, a single reel of magnetic tape is run through two tape recorders, one recording and one playing back. The sound recorded by the first device degrades over time as it is recorded over by the second machine, which plays back the ever changing tape. Here, the tape recorders are not just tools, but their properties are being used in the same manner as instruments.

Moving away from the magnetic tape tradition, Robert Ashley's *The Wolfman* constitutes one of the first uses of acoustic feedback in contemporary classical art. Ashley's work requires the performer to set their microphone's gain to a level just below the level at which it would feed back. When the performer sings into it, the feedback loop creates a howling effect, which Ashley notes is influenced by the acoustic treatment of the room. Steve Reich's *Pendulum Music* is another example of feedback work, where four microphones are swung over four loudspeakers. The microphones and speakers feed into each other, creating a feedback tone when they pass. As the microphones swing to a stop, the feedback becomes longer and more overlapping, until all four signals are still and unchanging.

For a more modern example, vocalist and electronic musician Lesley Flanigan employs custom-made feedback instruments. She manipulates sounds like feedback and electronic crackling to support her voice, which she often loops, distorts, and alters. Says Flanigan: "I was looking at a screen instead of listening to sound. It was a sterile, isolating experience, based entirely on recording rather than performing. I love electronic music and wanted to work with electronic sound, but I also really wanted the otherworldliness of electronic sound to be infused with the rich, raw, and almost dirty sense of live sound. I just couldn't figure out how to do it in a meaningful way".[33]

[33] Miller, Tyler. "The Speaker Is Present: A Conversation with Lesley Flanigan." IMPOSE Magazine, March 29, 2016. https://imposemagazine.com/features/lesley-flanigan-hedera-interview.

All of the sounds employed in the above examples could be reproduced by other (possibly more conventional and convenient) means, but they are ultimately made unique by the mediums in which they are generated. Were Reich's *Pendulum Music* presented as digital playback, it would lose some of its aesthetic appeal; *Pendulum Music* is not unique for the sounds that it makes, but how it makes them. This line of thinking might be familiar to anyone who finds vinyl records appealing. It represents a physical, tangible, and real-world reproduction of a sound, whereas digital playback is a facsimile of that sound.

The Space in Between

By identifying the various functions electronic elements can serve in music and the impacts they have on performance, we as performers can become better equipped to adequately integrate electroacoustic practices into our lexicon. As with all categorical systems, though, no truly definitive divisions can be drawn that accounts for all possibilities. A great number of works of electronic music fit into more than one of the categories proposed here and it is becoming increasingly commonplace to have works that transcend these categories by introducing multimedia elements that further complicate things.

Refer back to Volness' *Particle Waves*, whose first movement we have already identified as category two on the interactivity spectrum. The attaca second movement, however, opens up into semi-aleatory and introduces a fixed media track that the performer must reference for their place in the piece. After exerting a large degree of creative control over the trajectory of the piece with digital delay and reverb supplying an extra supportive dimension, the fixed media enters and we are suddenly jolted from a degree 2 work to a degree 1. The relative freedom of the first movement juxtaposed with the performer's newfound attentiveness to the track, drawing a unique narrative only possible in music with nonhuman media.

The vagueness of these categories is made more fascinating when digital means overlap with physical electronic works. Consider the work of Tristan Perich, whose primary output features music written for acoustic instruments and 1-Bit electronics. Perich does not employ any DAW or VST for his work, but codes the electronics manually, processed on custom computers and played back on custom speaker cones. The player even keeps track of their place in the piece using a specially-built LED monitor that displays the current measure number, and volume is controlled not by a dial or fader, but by the amount of voltage sent to the speaker.

Perich presents a dilemma. At a glance, his work seems to be a hard category one. The sounds of the performer are not altered in any way, the electronics do not react or change depending on any input from the player or otherwise, and the performer is at the mercy of the click track displayed on the LED. The problem with placing Perich's work in degree 1 (uninteractive) is that the electronics are not, technically speaking, playback. Perich's electronic media are not audio files saved on a hard drive, but are programs running in real time during a performance. So while the performer is locked to the electronics, the electronics are not a passive player like they would be if it were a digital playback situation. This places Perich more in the physical electronic music category despite his digital and unchanging electronics.

The Effect Spectrum

Now that we've established a system to classify the ways in which we interact with electronic music making elements, we need to do the same with the effects that these elements have on sounds. A considerable amount of scholarship already exists on this subject, the most fundamental of which being the music and writings of Schaeffer. For our purposes, the **effect spectrum** is a scale to quantify the types of alteration that are applied to sounds by electronic means. Much like we did with interactivity, we will measure effect in several degrees with the addition of a category that does not fit neatly onto the linear spectrum.

Degree 1: Unaffected Music

Unaffected music refers to any electronic media where acoustic sound sources remain unchanged. If a composer has made use of glass breaking, it sounds like glass breaking. Referencing the existing terminology, this could include acousmatic music. The sounds have been stripped from their sources and played back over loudspeakers, so the listener's concept of where the sound originated is blurred.

This degree of the effect spectrum is more transparent and accessible than music that has been more aggressively altered. An excellent example of this is Steve Reich's *Different Trains*[34], for string quartet and fixed media. Reich samples sirens, bells, and whistles associated with trains, and human voices describing routes that those trains might take (the drama of the work is derived from the juxtaposition between American passenger trains and trains of Nazi Germany used to transport prisoners to

[34] Reich, Steve. *Different Trains*. New York, New York: Boosey & Hawkes, 1988.

concentration camps). The piece is remarkably literal, and the only obvious processing is pitch shifting whistles to create melodic content.

EIP: *Conversation*

Unaffected music is best explored by returning to *Conversation* by Caleb Evans, specifically to the vocal samples in the fixed media. The first iterations of the sample are presented without processing to lend a sense of clarity to the audience. As the piece progresses, the same sample is run through a vocoder in order to meld the voice with the chords in the synthesizers. By first presenting the voice unaffected, Evans sets the listener up to comprehend the electronic media more completely.

Degree 2: Affected Music

Our second effect degree consists of music that has been processed or modified, but the instrument or source material is still identifiable. This often goes hand in hand with the second degree of interactivity, altered sound. Returning to our pedalboard examples from earlier, a distorted guitar signal is still identifiable as a guitar (in some musical contexts, it might even constitute the most familiar guitar sound possible). Basic pitch shifting, reverb, and other common effects also fall into this category.

The defining characteristic of affected music is the audience's understanding of the source material. In Martin Matalon's *Traces II*[35] for viola and live electronics, the computer records samples of the performer on stage in real time and plays them back at altered pitches and speeds. Because the audience has already heard the electronic

[35] Matalon, Martin. *Traces II*. Paris, France: Éditions Billaudot, 2005.

source material being played by the performer, they are likely to make the connection that the two are related. Comparably, Jonathan Andrew Smith's *Inquietude*[36] for kalimba and live electronics employs a variety of audio effects, from delay units to pitch shifting, but never goes so far as to obscure the characteristics of the instrument. An audience is likely to relate most or all of the electronic sounds to the kalimba.

EIP: *Particle Wave*

Movement 1 of *Particle Wave* by Kirsten Volness is an excellent example of affected music. The only electronic elements present in the movement are simple delay and reverb effects. While it is always apparent to the audience when electronics are being used, it is never unclear where those sounds come from. When either effect is switched on, it is understood that the sound is still coming from the vibraphone.

Degree 3: Modified Music

The final degree is modified music, or music where the source material has been somewhat or totally obscured by electronic processing. In this degree, sounds are changed to the point that a listener might not identify their source correctly. An excellent example of modified music is Dan VanHassel's *fzzl* for snare drum and live electronics. VanHassel's patch contains short samples from artists including Jimi Hendrix, Weezer, and Ravi Shenkar. A contact microphone is connected to the drum, and each time it is struck a sample is processed and played back. During the second half of the piece, samples are granulated and sustained to create a bed of sound that the percussionist

[36] Smith, Jonathan Andrew. *Inquietude,* Self Published, 2020.

performs over. The sounds in *fzzl* are processed to the point that an audience member cannot feasibly identify them during a performance of the piece. The only way to identify what is being fed into the patch is to listen to the stems themselves outside of the context of the performance.

Many works by composer Joo Won Park can be categorized as modified music, but with a twist that challenges our understanding of the spectrum. *Toccata* (2009) is a work that features objects (often items like slinkies and wind-up toys) played on a wooden board that is connected to a contact microphone running into SuperCollider. The soft and unassuming sounds of the objects are greatly amplified by the contact mic, and are processed by a variety of difficult to discern systems on the computer. The result of this process is a massive, chaotic, and noisy performance featuring sounds that an audience would most likely not have previously associated with the objects producing them. The twist is that Park's piece is intended for live performance, meaning that both his actions and objects are entirely visible to the audience. This means that an audience is met with clear sound sources, but the sounds they are experiencing do not necessarily line up with what they see.

EIP: *Inquietude*

Jonathan Andrew Smith's *Inquietude* for solo kalimba and electronics is an excellent example of how modified music might function. Much like Evans' *Conversation*, Smith first presents the sounds of the kalimba more or less clearly with a simple delay, placing it in the affected music category. As the work progresses, though, Smith's processing mechanisms become increasingly complex until the electronic portions of the sound are no longer distinguishable as coming from a kalimba. This escalation leads to the climax of the work, where the instrument sounds more like an organ than a kalimba.

The Other Axis: Synthesis

Just like the interactivity spectrum, the effect spectrum does not account for the entirety of the electronic music world. In this case, there is actually a considerably large subset of music left out of the conversation: synthesized sound.

Synthesis presents an interesting dilemma to our efforts to classify audio processing effects, mainly because synthesized sounds are not necessarily processed in the same way real world sounds and recordings are. Synthesized sounds are generated either by specialized analog devices or by digital means, and can either be employed to create entirely unique sounds or to emulate sounds that exist in the real world (this is referred to as “physical modeling”). The problem with classifying synthesized sound on the effect spectrum is that they are not affected by electronic systems, but are quite literally born into them. This is perhaps the most difficult part of electronic classification, and can easily be subject to technicality. For example, whether a work employs a sample of a real snare drum or a synthesized snare sound can completely change how it is classified on our scale. This begs the question: why do we care?

It is easy to assume as contemporary classical musicians that most synthesized instruments are developed to emulate “real” instruments, and are a poor substitute that is chosen because of constraints in budget, time, or availability. This narrative might have made sense at one point, when synthesized instruments were employed because of technical constraints (the 8-Bit sounds of early video game music were chosen largely because they could be produced by the primitive computers running these games). However, synthesized instruments have been available for long enough that generations of listeners have developed an affinity for their distinct timbres. They are now more likely to be used as a matter of aesthetic choice rather than of an economy of means.

A familiar example of synthesis as aesthetic choice might be the famous 808 drum machine. Originally produced by the Roland company between 1980 and 1983, the 808 marked one of the first attempts at a programmable drum machine that generated its own analog sounds. It was chided by reviewers on release for sounding unrealistic. As the decade progressed, the 808 found a market in popular music precisely because of its unique synthesized sounds and its relative affordability. It would go on to become one of the most used drum machines in popular dance music. While its synthesized timbre might have originally been a detriment and its cheap price point might have influenced its rise, the 808 has remained in the popular musical consciousness because it produces a unique and desirable aesthetic.

Take Matthew Joseph Payne's *Flight of the Bleeper Bird*[37] for solo flute and Gameboy. Here, Payne manipulates the four-channel sound card in the Nintendo Gameboy system to playback a predetermined sequence and treats the flute soloist as the fifth voice in the texture. It goes without saying that the composer had more sophisticated means available to generate his electronic sounds, making his choice to employ the Gameboy sound card a deliberate aesthetic choice.

---

[37] Payne, Matthew Joseph. *Flight of the Bleeper Bird,* Self Published, 2013.

The Space in Between

Just like our interactivity spectrum, the effect spectrum is not a monolithic system, and most works qualify for more than one category. This overlap is perhaps even more pronounced in regards to effect because of the dramatic capabilities of electronically processed sounds. By occupying different places on the spectrum at different points, a work can establish a dramatic trajectory comparable in power to form.

In Brian Ellis' *My Battery is Low and it is Getting Dark*[38], the work begins with the performer alone, recording different motivic cells to be altered and played back. When the first cell is faded in, it is fairly clear that the sounds being produced are directly related to the actions of the performer, and the audience develops an understanding of the process in play. More samples are recorded, changed, and added to the texture as the work progresses, eventually resulting in a dense wall of sound that continues to evolve over the duration of the piece. Because of this gradual shift, the audience begins with a clear idea of what the sounds are, but over time might begin to hear the electronics as increasingly independent of the performer.

It's About Sound

By defining two spectra to classify contemporary classical works with electronic elements, we as musicians are able to better articulate the performative implications of integrating electronic media into our traditionally human-centric value system. The interactivity spectrum establishes how a performer interacts with electronic media, while the effect spectrum details what these electronic elements are actually doing. While it is

[38] Ellis 2019.

perfectly possible to perform music with electronic media without these definitions in place, they afford a better understanding of what we are doing as performers when we engage with electronic media. Further, they help us to (broadly) predict how audiences will react to works with electronic media.

During a discussion surrounding her contribution to this project, Elainie Lillios gave an appropriately concise quote regarding the nature of electronic composition: “It’s about sound”. We judge a vast majority of western art music on form, harmony, melody, and other established theoretical bases. Because of the seemingly unlimited sonic capabilities of electronic musical tools and electronic music’s historical grounding in the experimental, it is often better to judge electronic music on different criteria, the most prominent of which being timbre.

Part 2: Practice

Using this Section

Like musical performance, working with audio technology tools is an endeavor without a single “correct” set of rules. Any implementation of a compressor, equalizer, or audio processing language that achieves a desired outcome is appropriate. However, there are certain practices that can save a user’s time and energy, or elevate the product from passable to excellent. All instruction that follows is suggestive, and I encourage you to experiment with settings and configurations outside of my instructions to either discover why I make the suggestions that I do, or to find a better solution than I have offered.

Along these same lines, there is a massive variety of tools that a performer might encounter that accomplish similar functions, from DAWs to microphones. While I make some references to specific products and a few deep dives into the most common proprietary tools in live signal processing, I have opted to keep most explanations here as broad as possible. For this reason, a certain level of experimentation and trial-and-error is suggested when using this resource. Almost all concepts discussed here are theoretical rather than literal (save for the sections on Max/MSP and SuperCollider), so having your chosen tools available when working through this section is recommended in order to apply what is discussed.

Because of the broad nature of this text, a performer may find that a certain topic piques their interest or that there is not sufficient instruction to achieve a specific desired goal. For readers finding themselves in this position, my first suggestion is to locate the documentation of the hardware or software in question. One of the saving graces of working with technology is that the engineers behind it keep a careful record of how everything works. Some of this information can be found with a simple Google search, but often more detailed information for software can be somewhat buried. Appendix II covers accessing the help files for some of the most common programs in the field.

Audio Technology Basics

Of all the various barriers to entry to the musician seeking to work with electronics, terminology might be the most intimidating of them all. While most performers are well versed in the jargon we use in creative settings (pitch, dynamic, phrase) the comparatively clinical vocabulary of audio technology can feel daunting. In this chapter we will cover the most prevalent and necessary terms and systems used in recording, playback, and live processing situations to give the performer a broad sense of how these systems function.

Digital Audio

While audio technology was once relegated to specialized (and expensive) sound generating devices, most of the electronic music a performer will interact with today is created, edited, and performed on a personal computer. This means that most of the electronic sound we are hearing in performances or during playback is digital as opposed to analog. These two terms are often the subject of value statements regarding fidelity ("music sounds better on analog equipment", or "I prefer to listen to vinyl records"), but in reality they are nothing more than two ways to send a signal from A to B, and both have pros and cons.

- An analog signal consists of a continuous signal at a variety of voltages that represents the vibrations captured by the microphone or pickup. This is how sounds work in the real world.
- A digital signal is one that has been encoded to only include two discrete voltages: 0 (off) and 1 (on). A single unit of this kind is called a bit, which is shorthand for binary digit. Digital audio consists of a series of bits derived from an analog recording system using a device called an Analog-To-Digital Converter (ADC). The inverse of this device is employed during playback from a digital system, called the Digital-to-Analog Converter (DAC).

With these definitions in mind, it is clear why digital audio has a reputation of being of a lower quality than analog. Figure 1 draws this issue into clear focus. While the general shape of our theoretical sound wave is preserved in its digital form, the sharp ridges caused by the ADC make it hard to believe that digital audio could ever even come close to its analog counterpart.

To understand how digital audio can reach a level of quality that is comparable to an analog signal, the concepts of bit depth and sample rate need to be defined. These two terms are often encountered when generating a project in a DAW or audio programming language, and influence the level of perceived realism in digital audio.

The simpler of the two terms is sample rate which, appropriately, can be defined as the rate at which a computer samples audio. Remember that a digital signal is discrete rather than continuous, so audio cannot be stored as one big sample. Instead, computers store digital audio in miniscule samples of sound which are played back quickly enough that the human ear cannot detect breaks and the signal sounds continuous. For example, the standard sample rate for compact discs (CDs) is 44,100 samples per second. This means that every second, 44,100 individual samples are referenced to create the illusion of a continuous analog sound. Higher sample rates like 48,000 or 96,000 give a better representation of an audio signal because they are more dense. It is also common to encounter sample rates measured in kilohertz (kHz), so the common sample rates of 44,100/48,000/96,000 Hz can also be written as 44.1/48/96 kHz.[39]

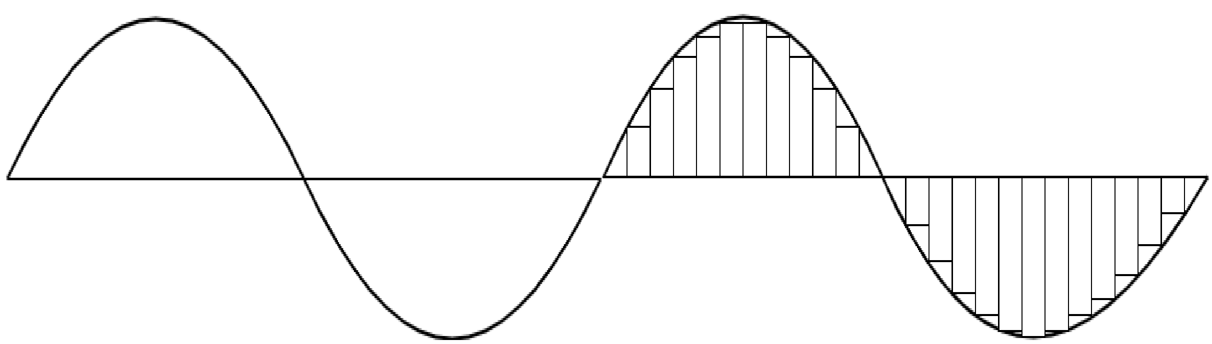

Figure 1. Representations of analog (left) and digital (right) sine waves.

The question must be asked: why 44,100 samples per second? The number sounds relatively arbitrary, but is justified by the acoustic phenomenon known as the Nyquist Theorem, which states that for a digital signal to be perceived as realistic, it must have a sample rate of twice the highest frequency in that signal. So for a digital recording of a sound whose highest frequency is 1000 Hz to sound realistic, the sampling rate must be at least 2000 samples per second, or two kilohertz. The highest frequency that a human can perceive is roughly 20,000 Hz, so a sampling rate of 44,100 accounts for slightly more than the human hearing spectrum. At just over double

[39] Roads 26.

the highest sound a human can hear, a sampling rate of 44,100 allows any sound a human could perceive to be digitally reproduced accurately.

Bit Depth refers to the number of bits in each sample of audio, and primarily affects the dynamic range and realism of a digital signal. Higher bit depths result in higher dynamic resolution because they allow for a far more detailed realization of amplitude. While knowing the math of bit depth is not strictly speaking necessary to performers, understanding how it works is helpful in comprehending why it is so important and picking the appropriate bit depth for your project. For the performer without interest in the more granular details, know that the CD standard bit depth is 16, with higher fidelity recordings using 24 or 32.

Remember that *bits* are binary units of information, and can send two different signals (0 or 1). So if we have one bit, we can represent two levels of amplitude by either sending a 0 or a 1. If we have two bits, we have increased the number of possible combinations to 4 (00, 01, 10, and 11), thus allowing for 4 levels of amplitude. The number of possible integers/amplitudes at a given bit depth can be found with the equation 2^X, with X representing the number of bits.

Bits	Possible Integers/Amplitudes
1	2
2	4
4	16
8	256
16	65,536
24	16,777,216
32	4,294,967,296

Table 1. The number of possible integers / amplitudes available at different bit rates.

On Table 1, we see that increasing the bit depth dramatically increases the number of amplitude levels available in digital audio.

For real-world context, consider the 8-bit video game music of the 1980's against modern popular music at the CD standard bit depth of 16. Despite merely doubling the depth, this increases the resolution of amplitude by a factor of 256, giving 16-bit recordings a far more realistic sound than their 8-bit counterparts. For a creative use of bit depth we can reference the music of Tristan Perich, whose output consists primarily of music written with 1-bit electronics. Perich's *1-Bit Symphony*[40] is generated entirely from electrical signals sending either an on (1) or off (0) message.

Microphones

If a performer wants to work with digital audio it is likely that they will have to employ some kind of microphone to record, amplify, or process their sound. They come in numerous shapes and sizes, from relatively affordable portable recorders to expensive single-purpose condenser microphones. While it is not necessary for the performer to know the inner workings of every type of microphone, awareness of the general varieties available and how they might interact with a given instrument can be vital in both the studio and home recording situation.

There are three primary categories of microphone that the modern performer might encounter:

- Dynamic: Dynamic microphones are most often found in live settings because of their relatively low sensitivity and high durability. The windscreen contains a conductive metal coil wrapped around diaphragm, which is suspended inside of an electromagnetic field. When a soundwave comes into contact with the coil, it vibrates across the magnetic field, generating an audio signal.
- Condenser: Condenser microphones are primarily found in recording situations, and are more delicate and sensitive than the average dynamic

[40] Perich, Tristan. *1-Bit Symphony,* Physical Editions, 2010.

microphone. They are also distinct from dynamics in that they operate at a lower voltage, and require an external power supply to function. Typically, this is accomplished by employing what is called Phantom Power over a preamp, usually 48 volts. Rather than a coil around a diaphragm, condensers contain a fixed metal plate and a moving diaphragm which together make a capacitor (known in the UK as a condenser, thus the name). The capacitor holds an electrical charge, the capacity of which is changed when a sound moves the diaphragm. This runs through a resistor to become the audio signal of the microphone.

- Ribbon: Ribbon microphones are perhaps the least common configuration that a performer might encounter, as they rarely make an appearance in live sound settings. Rather than a diaphragm and metal coil, ribbon mics contain a thin metal ribbon suspended between two magnetic poles. They often feature a figure-eight polar pattern, and their frequency response closely resembles that of the human ear. This makes them highly desirable for high fidelity recording.[41]

[41] Huber, 110 - 116.

Figure 2. Examples of microphone choice in the EIP. James Parker's "Baptism of Wind and Waves" (left) is best treated with a small diaphragm condenser microphone to capture the shimmer of the bells. Caleb Evans' "Conversation" (right) features exclusively drums and is best recorded with dynamic microphones.

It would be easy to assume that microphones listen more or less in one direction. Most depictions in popular culture reinforce this assumption. While it's not entirely incorrect, the reality is slightly more complicated than that. A microphone's polar pattern refers to the direction(s) that it listens in. There are several patterns that a microphone might have, including the following:

- Cardioid: This is the most common polar pattern in modern microphones, and is named for its vaguely heart-like shape. A cardioid microphone is most sensitive at the front and least sensitive at the back.
- Supercardioid: A more directional version of the cardioid pattern. These microphones are still most sensitive at their fronts, but reject slightly more side noise and are more responsive to sounds behind them.
- Hypercardioid: An even more directional variation on the cardioid pattern. This polar pattern looks similar to the supercardioid pattern but features more side rejection and an even more present rear response.

- Omnidirectional: These microphones are equally sensitive in all directions, making them great for well-conditioned rooms but problematic in untreated spaces. They are not ideal for live sound situations since they do not reject sound coming from any particular direction, and for this reason they are prone to feedback.
- Bidirectional: This polar pattern is, like cardioid, named for its appearance. Bidirectional microphones have the least response of any pattern at their sides, and are equally sensitive at their front and back. This makes them particularly equipped at achieving an isolated signal in a situation with a variety of sounds (chamber music setting, rock band, etc). The standard polar pattern for ribbon microphones is bidirectional.
- Shotgun: Microphones featuring this pattern are most common in film and television settings. Extremely directional, shotgun microphones reject most sound behind them and to their sides, making them great for difficult recording environments (like, for example, live on-site television interviews). In recent years, shotgun microphones have been introduced as sound reinforcement for marching bands and drum corps.[42]

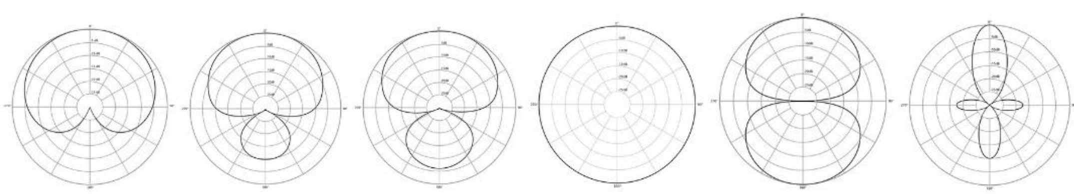

Figure 3. Left to right: cardioid, supercardioid, hypercardioid, omnidirectional, bidirectional, and shotgun polar patterns

[42] Huber, 116-120.

Polar pattern marks one of the most important technical distinctions between microphone models and are useful choosing a microphone for a given application. The other vital technical specification to understand when choosing a microphone is its frequency response.

Microphones all have unique frequency response ranges that vary by model. While most modern models have a coverage for the complete range of human hearing (20 Hz to 20,000 Hz), they often have different responses across this spectrum that affect their sonic quality. The classic Shure SM57 is a microphone designed for general instrumental use. Its frequency response is somewhat flat, with a minor bump around 4,000 - 5,000 Hz to capture the high overtones of an instrument. On the other hand, the Shure Beta 52 is a microphone designed specifically for use on kick and bass drums, and has an appropriately severe frequency response designed to capture the low frequencies and attacks on these instruments.

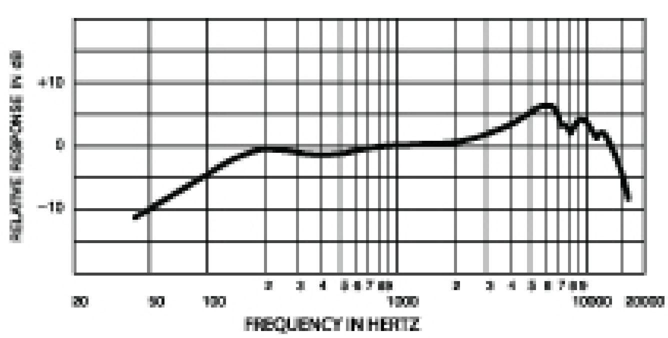

Figure 4. The frequency response curve for the famous Shure SM57.

Audio Connectors & Interfaces

To properly route a signal to its destination, a few common connectors are employed.

- XLR: The standard connector for microphones. XLR connectors feature three pins to communicate a signal. The “female” end of the cable receives the signal from the microphone, and the “male” end relays it to whatever is next in the signal chain.[43]
- TS/TRS: Often called *quarter-inch cables* in reference to their size, these two connector types are often mistaken as identical. TS stands for “tip sleeve”, referencing the tip and the long metal sleeve that makes it identifiable. TRS stands for “tip ring sleeve” and looks exactly like a TS connector save for an extra ring around the sleeve.
- RCA: These connectors were originally designed by the Radio Corporation of America to connect turntables and radio receivers. They are recognizable by their color-coding system, with white being left audio and red being right audio (yellow connectors are also common and carry a video signal). A fairly common subset of RCA called S/PDIF (Sony/Philips Digital Interface) is present in many modern audio interfaces but is rarely used.
- Speakon: A unique line level connector meant to carry signals over long distances. Speakon connectors are primarily used to connect to speakers and

[43] Rayburn, Ray A. XLR History. Sound First, December 20, 2017. http://www.soundfirst.com/xlr.html.

stage monitors, and feature a unique locking mechanism to avoid getting mistakenly pulled out.

- MIDI: MIDI (Musical Instrument Digital Interface) cables are less common now that USB technology has become fast and common enough to carry MIDI data, but they still make appearances in older systems and in specialized analog synthesis units. MIDI connectors look similar to XLR, but feature five pins instead of three. These connectors are also unique in that they require two linked connections to function: a send and a receive. This is why MIDI inputs always appear in pairs on hardware.
- USB: Standing for Universal Serial Bus, USB is not a design protocol unique to audio equipment, but a connection design employed by almost all modern personal computers. USB connectors allow computers to interface with other devices including cameras, hard drives, audio interfaces, and even with other computers. They come in several varieties that can all interface with each other, the most common types being USB A and C. The ubiquity of USB has led audio technology companies to develop microphones that connect directly to a computer's USB slot. While not suitable for professional recording or performance scenarios, audio devices using USB tend to be cheaper and easier to use than their XLR or TRS counterparts, and are well-suited for low-stakes, personal projects.

While each of these connectors is considered standard for their functions, only USB is directly compatible with personal computers. To record or real-time process an audio signal with the more standard connection types, it is necessary to employ an audio interface.

Figure 5. The most common connectors a performer might encounter. From top left to bottom right: XLR, TS, TRS, RCA, Speakon, MIDI, USB.

An audio interface is a device that translates audio signals from their natural analog states to a digital signal that a computer can read and make changes to. Most interfaces feature inputs and outputs for receiving and sending signals, and employ several connector types to account for the assortment of devices it is required to interact with. Inputs tend to receive signals over either XLR or TRS connections, main outputs primarily use TRS connections, and there are often additional receivers for uncommon or dated connectors like MIDI, RCA, and S/PDIF.

When choosing an interface to record or perform with, there are a few technical considerations to make. The first is the size of the device, judged by the number of inputs and outputs. For a simple home setup requiring a single microphone and playback, something with one input and two outputs will suffice; remember that standard audio playback is in stereo, so you will encounter at least two outputs on almost any interface. But more complicated projects might require a larger interface with a greater number of ins and outs. For instance, performing a piece in four or eight channel surround sound will require that number of outputs to route tracks to.

Additionally, be aware that it is fairly normal to encounter expandable interfaces. These interfaces feature an optical ADAT connection that receives a second hardware device that expands the in / out capabilities of the interface to its advertised specification. For example, take the second generation Scarlett 18i8. The Scarlett naming scheme lists the number of inputs followed by the number of outputs, leading the user to believe that this model has eighteen inputs and eight outputs. However, there are only a total of eight functional TRS or XLR inputs on the device (four across the front of the device and four line inputs on the back), and only what look like four outputs (two main outputs and two headphone outputs). Upon further reading, the user finds that the 18i8 name comers from counting the interface's rarely useful S/PDIF connectors as two input and two outputs, the ADAT expansion's eight additional ins, and that the headphone outputs actually send in stereo, making them slightly more difficult (but possible) to route. The 18i8 is still a capable and highly useful interface that nonetheless offers a cautionary tale in checking technical specifications.

Digital Audio Workstations (DAWs)

Digital Audio Workstations, or DAWs, are software applications used to create, record, mix, and play back audio files. If you have ever taken a course in audio technology or spent any time in a recording studio, this is the software you likely encountered. Most DAWs are designed for popular music production, but generate any variety of audio content, such as dialogue, Foley sound, radio work, podcasting, classical music, and more.

Most DAWs the working musician will encounter are nonlinear editing systems, meaning that they are nondestructive: making an edit does not make alterations to the original data. Further, edits are not locked to the order of events of the file, and can be made to whole sections of audio at once. This contrasts a linear editing system where a user would be required to scrub through the data in order to make changes, like with older magnetic tape systems.

DAWs operate in channels, with one channel for every individual signal. Often, a DAW will group stereo signals into one channel, but give the user the option of separating them. A user can move audio or MIDI data around to change how it plays in time. Additionally, users can make changes to the qualities of audio (timbre, loudness, etc) to balance and improve a project. This process is referred to as *mixing*.

There are a wide variety of DAWs available to the modern musician. While most of them have a similar workflow, there are differences in their interfaces and workflows that might influence a prospective user's choice. What follows is by no means an exhaustive list, but references the most common tools that you might encounter.

- **Pro Tools**: Often considered the industry standard DAW by recording engineers, Pro Tools by Avid excels in the recording and editing of live audio. While perfectly capable of handling MIDI data and digital instrument input, its interface is less adept at handling this kind of information, making for a clunky workflow. Pro Tools is recommended for the user who is primarily interested in recording and editing what we might consider "traditional" popular and classical music.

- **Ableton Live**: A popular program among producers of popular electronic music, Live might be the most unique tool listed here. While it operates on the same basic logic as most other DAWs here, Live lives up to its name by including an interface to allow for the real-time loop triggering, making it possible to perform music written in MIDI live. The inclusion of this feature introduces a number of quirks to the workflow that gives Live its distinct feel. Perhaps the antithesis to Pro Tools, Ableton Live's workflow favors synthesis and digital instruments while working with live audio is slightly less intuitive.
- **Logic Pro**: Apple's Logic Pro is a Mac-exclusive DAW that many consider the most user-friendly of the common tools. A fairly balanced program, Logic is more or less equally equipped to handle both live audio and digital instruments. The built in synthesis tools are notably robust, making it a favorite of electroacoustic composers. Logic's weakness is in user interface: what one gains in user friendliness is lost in detail-oriented editing, and the experienced user might find Logic to be clumsier than its kin.
- **Reaper**: Reaper is a relative newcomer to digital audio, with its initial release in 2006. Something of a miracle in digital audio, Reaper is the most affordable paid DAW available and features the lowest processing requirements. Less user-friendly than its more expensive and robust

counterparts, Reaper's learning curve is certainly steeper than those of Logic, Pro Tools, etc).

- **Cuebase**: A slightly less fashionable DAW, Cuebase is a less common but capable program available on both Mac and PC systems. Cuebase offers a fairly balanced experience between digital instruments and live audio. On the downside, its interface is menu-heavy, which can make a first-time user's experience arduous.
- **FL Studio**: Originally simply titled FruityLoops, this DAW is an excellent tool for popular electronic-music making. It features full live audio recording capabilities, but its live audio interface is less intuitive than its MIDI interface.
- **Reason**: This DAW visualizes all of its instruments, plugins, and effects as rack units, allowing the user to physically route audio to better understand signal flow. Reason's primary focus remained on MIDI control and synthesis until 2009 when live audio recording capabilities were introduced.

The Environment

Since there is much information to account for, DAWs tend to have several display modes that a user can toggle between, depending on what they are trying to accomplish. The two primary displays a user will encounter in their DAW of choice are the Mix view and Arrangement view. A system like Pro Tools splits these two views into separate windows that a user can resize or minimize to clear up space. Conversely, Ableton Live keeps everything in one window but maps a key to allow quick toggling between the two views.

The arrangement view displays a user’s project in time. A project progresses left to right and channels are stacked vertically to show different signals; each has a lane where audio can be moved around. Audio is displayed in blocks in these lanes and can be spliced, moved, faded, or any other number of adjustments to change how they play against each other. The arrangement view also includes a playhead, a moving vertical line that shows where in your project playback is occurring.

The mix view displays all of the information one might find on a physical mixing console, including faders, gain control, pan, mute, and routing. Each channel corresponds directly to its similarly named channel in the arrangement view, and one affects the other. This is also normally where a user finds places, or independent programs that are housed inside your DAW.[44]

Routing

Routing, also referred to as signal flow, is a fundamental concept to almost all endeavors in electronic music-making, and refers to the path (route) a signal takes from its inception to its destination. The ability to track a signal and define what is happening to it at different stages of its journey is necessary for troubling shooting playback systems, and is an incredibly helpful skill for the aspiring electronic musician. While the term routing is applicable to all forms of electronic music-making, we will be discussing

[44] Jones, Andy. “The Beginner's Guide To DAWs - The Basics.” MusicTech, June 21, 2017. https://www.musictech.net/guides/essential-guide/essential-guide-daws/.

it here as it relates to DAWs since they are the most frequently encountered form of the practice.

To assess routing is to locate inputs (where sound enters) and outputs (where sound moves forward to its next destination). Each channel in a given DAW has both an in and an out, and can be manipulated to take a signal from and send a signal to any other channel. You will also find inputs and outputs on Mixers, Interfaces, Direct Input boxes, and any other device meant to process, alter, or route a signal. When a user clicks an input for a channel, the DAW will display all of the signal sources it is aware of. Similarly, it will display all of the destinations it is aware of when a user clicks a channel's output. By default, most DAWs will look to your computer's built-in microphone for input and its built-in speakers for output.

This is helpful, but ultimately restrictive, for performance since they will rarely (if ever) play electronic media over laptop speakers on stage. To make use of the external devices that are necessary for this kind of performance, we need to tell the DAW where to look for input and output sources. In almost all programs, this information can be found in some variant of the Audio Preferences page. Here we can choose input / output devices (sometimes referred to as a playback engine), which tells the DAW where to look for inputs and outputs.

The most simple example of routing is to take a signal in and send it out of the main outputs. Select the input you want to use for your microphone (this will more than likely be channel one) and the output you want to send it to (some variant of "1-2", "Master", "Main", et cetera). Congratulations! You've just sent a signal into your DAW, and back out again. What if we want to alter that sound in some way? For that, we will want to employ plugins.

In digital audio, a plugin is an independent program that is loaded and housed inside of a DAW that performs a specific task, like distortion, EQ, or compression. They are also sometimes referred to as VSTs (Virtual Studio Technology). Most DAWs come with a stock set of plugins that a user can employ, but there are thousands of external ones available either for free or purchase by companies like Blue Cat, Izotope, and Waves. Plugins can be applied to any channel in a DAW, and route in order from either top to bottom, or left to right.

The order of operations is vital when making changes to an audio signal with a plugin. This is because plugins run in the order they are employed. If a user places an EQ followed by a compressor on a channel, the compressor is altering a signal that is already equalized. If the order is reversed and the compressor is placed before the EQ, the EQ is altering a signal that is already compressed. These two orders of operation will have different outcomes ranging from the subtle to the notable.

Like all of the concepts discussed in this section, there is no ordained correct order to place plugins in. For simple mixes, it is generally suggested to compress before equalizing. The Los Angeles-based Icon Collective suggests this broad order of events:[45]

1. Subtractive EQ
2. Compression

[45] PQ, Rory. "What's the Best Effects Chain Order for Mixing?: Icon Collective." Icon Collective College of Music, May 4, 2020. https://iconcollective.edu/mixing-effects-chain-order/.

3. Additive EQ
4. Modulation Effects (Flanger, Tremolo, Chorus, et cetera)
5. Reverb and/or Delay
6. Limiter

In our discussions on routing within a DAW, we've sent a signal in and out, and made alterations to that signal by employing plugins. While this is enough information to tackle most needs for the amatuer electronic musician, it is just the tip of the iceberg regarding the power of most DAWs. Changing the ways in which we route signals can allow for a much greater control of a mix and save your computer's processing power by accomplishing repetitive tasks more efficiently. To do this, we'll need to employ a few extra types of channels.

Up to this point we have only worked with general audio channels in either mono or stereo, but there are several types of tracks a user might encounter in their chosen DAWs, including:

- Audio Channel: A channel that handles audio information. These are going to be the most numerous channels that get used in a mix.
- Master: The channel where all audio signals go before moving out of the DAW to speakers or headphones.
- Aux Channel: Short for auxiliary. These channels act as miniature Master faders, collecting signals chosen by the user into one place and sending them along to their destination (usually the Master). This allows a user to make collective changes to groups of signals that may need similar

treatment (for example, a group of like string instruments in a larger ensemble).

- Instrument / MIDI Channel: Channels designed to specifically handle MIDI instruments and VSTs. While they output an audio signal to whatever Master / Aux channel the user has chosen, these channels normally display MIDI information rather than audio waveforms.

While some of these channel types have relatively straightforward uses (MIDI channels are used to house MIDI instruments), there are some unique applications of other channel types, specifically the Aux channel. Aux channels act like miniature master faders, taking input from as many or as few audio channels as the user likes. This allows the user to develop a submix, which is effectively a mix within a mix. Rather than sending all audio channels to the master fader immediately, they can first be routed to an aux channel to give them a uniform mix.

As an exercise, say that we have a piece with a string quartet (two violins, viola, and cello) and a saxophone quartet. By sending the four string instruments to one aux channel and the four saxophones to another, a user is able to mix (EQ, compress, et cetera) all four instruments as one signal. This has the practical benefit of being much less CPU intensive than applying these plugins to each individual instrument, plus it lends the group of instruments a sense of sonic cohesion (it is fairly common to hear submixed compression referred to as the process that “glues a mix together”).

Next to submixing, aux channels are most often used to implement time-based effects like reverb and delay. The practical benefit of saving CPU is present here as well (on a mix with 20 tracks, it would be incredibly taxing for a computer to simultaneously run 20 reverb units), and it makes sense that you want the instruments in your mix to sound like they are performing in the same space. The more creative reason for keeping time-based effects on aux channels is an issue of control and cleanliness. While an effect like an EQ or compressor makes changes to quality of a sound, time-based effects fundamentally alter how long a sound will be active. For this reason, it is best to employ a technique called *parallel processing*.

Thus far, every example of signal flow that we've encountered has been what we might consider in-line processing. This style of processing takes place with one effect coming after another, each processing a signal that has been affected by the previous effect in the chain. This is the default way that DAWs set up plugins. Parallel processing is an alternative to this method, where a signal is sent to two locations: a dry (unprocessed) signal goes to the main out, while a second signal is sent to an aux channel for further processing. This is accomplished by using the sends available on a channel strip.

Sends can be thought of as optional extra outputs that route audio to locations other than their main out. If we want to route a signal to an aux channel as well as its main out, we would add a send to that channel that routes it to the input of the aux channel. For these purposes, most DAWs have internal routing systems called busses, used for sending signals to locations within the program. In a large project, we could route all of our audio to a single aux channel with a reverb plugin via sends. This way, all of the tracks receive the same reverb treatment and we've only had to employ a single plugin. Each send also has its own fader for volume control, meaning that one can alter the levels of reverb that each instrument receives.

Audio Effects

EQ

Equalization, often shortened to EQ, is the act of balancing (or equalizing) the different frequencies of a signal in order to achieve a desired sound. To accomplish this, an EQ unit deals with two factors of sound: frequency and amplitude.

Frequency is defined as the number of times an event occurs within a given period (for example, a heart rate or metronome measured in BPM could be considered a frequency), and can be thought of as a more detailed measurement of pitch. In audio, frequency is the number of times a wave completes its cycle and is measured in hertz (Hz), or cycles per second. Humans can hear roughly 20 to 20,000 Hz, which translates to 20 to 20,000 cycles per second. The higher the frequency, the higher the pitch we hear when that frequency is played. If a wave vibrates at a frequency below 20 Hz, the human ear stops hearing it as pitch, and starts hearing it as a rhythm.[46]

Amplitude, which can be likened to dynamics, is the measure of change in a wave during a period and is measured in decibels (dB). The higher the amplitude, the louder we perceive the sound to be. Since amplitude is a measure of loudness, it is not uncommon to see an amplitude control called gain. This term can mean several different things, but here it is simply a measure of loudness.

An EQ unit is usually visualized using a Cartesian coordinate graph, with the X coordinate representing frequency (Hz) and the Y coordinate representing amplitude

[46] Huber, David Miles, and Robert E. Runstein. *Modern Recording Techniques*. 8th ed. Burlington, MA: Focal Press, 2014, 47-55.

(dB). The user interacts with a number of nodes which allow them to alter the loudness of different frequencies. Each node controls a frequency band, or a range of frequencies surrounding it. Moving a node left or right will adjust the frequency up or down. Moving a node up or down will adjust to the amplitude to be louder or softer.

Each note also has a Q (standing for quality) control, which dictates the range of a frequency band. A high Q will result in a narrow frequency band, and consequently only affect a small part of the frequency spectrum. A lower Q will cover a much wider band, making broader changes to a signal's sound.

The frequency, gain, and Q are the primary ways a user makes changes to an audio signal in an EQ unit. But what if we want to cut out a certain frequency altogether, or make a uniform boost or dip after a certain frequency? To make use of the total power of an EQ unit, a user needs to understand the different filter shapes that a frequency band can have.

- Bell: The default filter shape of nodes of an EQ unit. The bell affects frequencies on either side of the node equally based on the Q, creating the visual shape for which it is named. Bells are primarily used to boost or cut specific frequency ranges on the spectrum.
- Low Shelf: Applies a uniform change resembling a shelf to all frequencies *below* a node. Most useful when a broad part of the spectrum needs a cut or a boost.
- High Shelf: Applies a uniform change resembling a shelf to all frequencies *above* a node. Most useful when a broad part of the spectrum needs a cut or a boost.

- Lowpass Filter: Cuts all frequencies *above* a given node. In a Lowpass filter, the Q affects the slope of the cut above the node, or how quickly the gain drops off. The name Lowpass refers to the fact that the filter passes all frequencies below the node.
- Highpass: Cuts all frequencies *below* a given node. In a Highpass filter, the Q affects the slope of the cut below the node, or how quickly the gain drops off. The name Highpass refers to the fact that the filter passes all frequencies above the node.[47]

While all of this terminology is helpful to understand what is happening in an EQ unit, using one to sculpt a signal can be a daunting task. Indeed, equalizing audio is a practice unto itself, and takes experience, patience, and experimentation to get good at. Much like musical performance practice, there are a set of theoretical guidelines that can be helpful to your first engagement with an EQ. These rules are made to be broken, though. Remember that audio engineering (which is functionally what we are doing here) is a results-driven exercise. If you break every guideline suggested below but the sound you have generated is desirable, you have achieved a quality EQ.

Before we begin, I like to stress the importance of subtlety in your EQ. Most dips and boosts to a given frequency band should be within 3-5 dB in the interest of preserving realism. This, like everything else in mixing / mastering practices, is a rule with many exceptions, but in general one should only start making drastic changes to an EQ curve when a problem cannot be solved with a subtle one.

[47] Huber, 484-489.

The first thing to consider when equalizing a signal is what that signal is and how it is recorded or produced. Is it an instrumental sound made in a controlled environment like a recording studio? Are you working with a vocal signal for a live band, where other instruments bleed into the microphone? It could even be a field recording, like the sounds of birds recorded in nature. These different inputs all require different touches and techniques to deal with, but the first question is always the same. What do I want it to sound like?

Remember that recordings generally include most, if not all, of the spectrum of human hearing, which is 20 - 20,000 Hz. For most musical instruments, a large portion of this spectrum is outside of the capabilities of that instrument. For example, a tenor trombone generally has a range of E2 to Bb4, about two and a half octaves. At A = 440 (the standard tuning for American orchestras and bands), E2 translates to 82.41 Hz and Bb4 to 466.16 Hz. This means that the tenor trombone cannot produce a pitch below 82.41 Hz, and we can remove that frequency range with a highpass filter.

Similarly, the trombone cannot produce notes above 466.16 Hz. Unlike our low signal, though, this does *not* mean that we can reduce or remove all frequencies above this point. Remember that instruments get their characteristic timbres from overtones, or a stack of simple waves at various amplitudes. While an A2 sounds the same on any given instrument, all instruments sound unique playing that note as a result of their variable overtone structures. If we reduce the overtones of an instrument, we will muffle its characteristic sound. This can be an inexact science in practice. Generate a lowpass filter at the high end of the frequency spectrum, and move it lower until you start to hear it affect the signal of the trombone. Once you reach that point, move the lowpass just high enough that the uninhibited instrumental sound comes through. By applying these filters, we have cleaned out many undesirable ambient sounds that might exist outside of the range of our instrument.

Now that we are dealing with only our desired instrumental signal, we can start thinking about how we might improve it. This is necessary because microphones are imperfect. Each has a unique frequency response that might not be ideal for a given instrument's (or performer's) unique tone. An understanding of frequency band characteristics will be helpful in identifying qualities of sound you want to accentuate or reduce.

Frequency Range	Characteristics
20 - 60 Hz	Extremely low rumbling sounds. Digital noise from bad equipment also shows up here often.
60 - 100 Hz	The useful low range of bass heavy instruments (bass guitar, tuba, etc). Boost to increase fullness and fundamental frequencies.
100 - 200 Hz	The low range of middle range instruments (Trumpet, French Horn, etc). Boost to increase fullness and fundamental frequencies.
200 - 300 Hz	The beginning/middle of the range of many vocalists. Is known to lend a muddy sound in some instruments, specifically the low rumble of cymbals.
400 - 800 Hz	Incredibly useful for adding fullness to middle range instruments since it covers the first harmonic in many cases. Tends to be undesirable on percussive instruments like piano and drums. Contributes to muddiness in many cases.
800 - 3,000 Hz	Sometimes referred to as the “telephone band”, this range lends a nasally, and somewhat fricative quality. Can be reduced for clarity in most cases, but can be useful for clarity on bass instruments.
3,000 - 6,000 Hz	Useful for boosting vocal presence and attacks in percussive instruments (drums, guitar, piano).
6,000 - 9,000 Hz	Useful for increasing the shimmer of most mid range instruments.
9,000 - 12,000 Hz	Houses the high harmonics of high range instruments, and can be useful to accentuate the shimmer of cymbals. Can be cut to make a signal darker.
12,000 - 16,000 Hz	Useful for vocal brightness.
16,000 - 20,000 Hz	Can be used to increase clarity if it’s an issue, but mostly contains hiss and unwanted room noise.

Table 2. General descriptions of different bands on the audible frequency spectrum.

While each instrument is unique and has its own quirks and qualities that require special attention, there are a few places on the spectrum that can act as a general starting place for developing an effective EQ curve.

- Fundamental Frequency: the first note in the overtone series, or the name of the note (for A = 440, the fundamental frequency is 440 Hz). This range coincides with the range of the instrument, making it relatively easy to identify. In most cases, accentuating the fundamental lends the sound with a fuller tone.
- Mid-Range: Appropriately, the middle range of an instrument's frequency spectrum. While vital to preserve the instrument's timbre, this range tends to be muddy, thick, and somewhat unpleasant. A small dip in an instrument's mid-range will usually lend a sense of clarity to a signal.
- High Overtones: The very high and often difficult to perceive part of an overtone series, generally speaking found between five thousand and seven thousand Hz. If clarity is an issue, making sure this range is adequately audible will ensure that a listener hears the grit and shimmer of an instrument.

In writing these suggestions, I have purposefully refrained from specifying frequency bands in order to keep this text applicable. Different instrument's fundamentals, mid frequencies, et cetera will fall in different places on the spectrum. These can be found either via experimentation or a quick Google search.

There are two primary principles of equalization, called additive and subtractive. Additive equalization involves increasing the level of frequencies that are useful, while subtractive EQ is the practice of reducing unpleasant frequencies to make room for the ones that are more desirable. Both have their place and a user can (and often does) employ both principles within the same EQ, but it is generally best practice to favor subtractive over additive. Digital audio is finite, and there is only so much signal to work with. When a user practices additive equalization they are effectively boosting a signal beyond what is naturally occurring, possibly lending it an artificial or damaged sound. On the other hand, Subtractive EQ carves away parts of a signal, which does not risk damaging the sound.

In more complicated mixes and in many mastering suites, it is not uncommon to see two EQ units on either side of the effect chain, sometimes referred to as pre-EQ and post-EQ. These allow the user to split their EQ into two parts, placing subtractive at the beginning and additive at the end. This way, the signal being sent to compressors, reverb units, etc. has been cleared of any undesired sound by the subtractive EQ. Once the signal has been fully processed, the additive EQ can be used to accentuate the frequencies that the user might feel need extra power.

Compression

Compression is a practice in audio mixing that is of equal importance and commonality to equalization, but is often largely misunderstood. This can be attributed to how difficult it can be to notice a well placed compressor, but the bigger culprit here is the term’s use in everyday digital culture. We talk about compression in streaming media and compress files on our computers to save space, but to compress audio is a fundamentally different process than anything we run into in our daily lives.

Audio compression is the process of reducing the dynamic contrast of a signal in the interest of making that signal more evenly perceptible on a wide variety of playback systems. Like EQ, compression is a response to the imperfection of recording and playback technology. While it can be viewed as reductive of the more dynamically variable input of the performer, it is the glue that makes a mix sound coherent, and is what allows a performance to sound “correct” when played back.

A compressor is made up of the following components:

- Threshold: A dynamic variable measured in decibels. When a signal is underneath the threshold, the compressor does nothing. When above the threshold, however the compress refers to the ratio.
- Ratio: a proportion represented by X:1, with X being the input of the user. When the threshold is passed by the signal, the ratio decides how much to reduce (or compress) the signal. *For every X decibels over the threshold, only 1 dB will be produced.* For example: If we have a 2:1 ratio and the signal is 4 dB over the threshold, only 2 dB will be produced.
- Makeup Gain: After the ratio has taken effect, we will be left with a softer signal. This allows the user to raise the overall level of the signal, making the quiet parts more audible.
- Attack Time: The amount of time it takes for a compressor to activate after crossing over the threshold, usually measured in milliseconds.
- Release Time: The amount of time it takes for a compressor to deactivate after crossing under the threshold, usually measured in milliseconds.
- Knee: The severity of the curve at the threshold.

The interconnectedness of these components is part of what makes compression so difficult to master. A poorly placed threshold combined with inappropriately quick attack / release times will cause the unit to kick on and off rapidly, lending a choppy feeling to the signal. If a ratio is set too high, the user might try to fix the problem by increasing the makeup gain, increasing the noise floor of the recording beyond its ideal level. While compression is one of the most powerful tools in perfecting a recording or amplified performance, it can take a great deal of experimentation to get a feel for it.

The threshold is the starting point for setting a compressor appropriately. Set it to a point where the compressor remains activated when the signal is live. While it is acceptable for the compressor to cut on and off, it should not do so constantly. Next, the ratio needs to be set. Because one of the primary concerns when working with contemporary classical music is preserving dynamic contrast, it is common for compression levels in these recordings to be relatively subtle (this is why orchestral recordings are often quiet when compared to popular music recordings). In general, I personally start from a ratio between 1:1 and 2:1. This helps to avoid overcompression, and keeps the level of makeup gain required modest. The loudest parts of the signal might be notably softer after setting an appropriate threshold and gain. At this point, the user can increase the gain (often referred to as makeup gain) to recover the lost loudness and increase the signal to an appropriate level. By following these steps, the user has reduced the loudest parts of the signal to allow its overall level to be increased, and made the softest parts of the signal more easily audible.

Reverb

Reverb might be the most widely acknowledged audio effect to the general public. While effects like EQ and compression are vital but subtle, most savvy listeners can quickly identify when reverb has been employed. As a creative audio effect, reverb takes its name from the acoustic phenomenon known as reverberation, which is the reflection of sound off of various surfaces causing it to continue past its initial activation. Hard, sheer, or dense surfaces reflect sound more efficiently than soft permeable materials. This is why a large concert hall with wooden floors is more reverberant than a small carpeted room.

Reverberation is present to varying degrees in our everyday lives, so it makes sense that it is an appealing quality for recorded audio to take on. Artificial reverb in small measure can lend a signal a full, realistic sound, while more heavy handed application of the effect can allow a musician to lend their signal a spacious quality, as if it was recorded in a concert hall.[48]

Reverb has historically been applied via several acoustic processes, the most straightforward of which is the use of an echo chamber. Here, an engineer would play and record a signal in a specially designed reverberant room. While realistic because it is, in fact, real, this method is expensive and difficult to make adjustments to. A more practical method of applying reverb is the use of a Plate reverb unit. Here, a transducer feeds a signal into a large suspended metal plate that is connected to a pickup. By running the signal through the plate, a shimmering reverberation is added to the sound.

---

[48] Huber, David Miles, and Robert E. Runstein. *Modern Recording Techniques*. 8th ed. Burlington, MA: Focal Press, 2014, 71-73.

While more practical than owning an echo chamber, classic plate reverb units were still expensive and unwieldy, some weighing as much as 400-600 lbs. A cheaper alternative to the plate method unit is spring reverb. Spring units operate in much the same way as plates, but rather than using a large sheet of metal they employ small springs to add the effect. These units are small and cheap enough to be fit into mobile casings, and are often found built into guitar amplifiers.

Plate and spring reverb units are still common, but it is more than likely that a contemporary classical performer will primarily encounter digital reverb. These units will often include presets and options like “Chamber”, “Hall”, and “Room” in addition to the classic plate and spring effects. These settings are meant to model the reverberant qualities of different spaces through a process called convolution.

Convolution is an operation in math where one function is introduced into another function, changing the original’s shape. In terms of audio, it refers to two signals being multiplied together to create a signal that contains properties of both. This has numerous applications in audio processing both regular and experimental, but convolution most often applies to reverb and is sometimes referred to as physical modeling.

In a convolution reverb unit, the input signal is multiplied by a recording of an impulse response in a physical space; an impulse being a short, loud, and percussive sound that activates the reverberant qualities of the room. When used in a convolution reverb unit, the impulse itself is removed, leaving only the room’s response to the sound. When an input signal is convoluted with the impulse response of a room, the signal is lent the reverberant qualities of that space. When a user selects “Hall” mode in their reverb unit, their input is being lent the qualities of a large hall's impulse response. This method has proliferated a number of reverb units specifically designed to emulate different famous recording and performance spaces.

Beyond the specific methods by which the effect is applied, there are several parameters that are present in most digital reverb units that a user should be familiar with.

- Room Size: The approximate size of the room the unit is emulating. While there are several methods of measuring room size in various reverb units, in general larger rooms tend to have longer decay, wider stereo images, and a more distant quality of sound. Smaller room size correlates to shorter decay, a more narrow stereo image, and a more direct sound.
- Early Reflections: Control over the early reflections of a sound off of a reflective surface. These reflections often sound more like an echo than reverberation, and can influence how close a sound feels to a listener.
- Pre-Delay: Usually measured in milliseconds, pre-delay simulates the amount of time it takes for a signal to reach a surface and make its first reflection. A larger pre-delay indicates a larger room and vice versa.
- Decay Time: The amount of time it takes for a reverberation to stop, usually measured in seconds or milliseconds. Some units will link decay time with room size in to help users avoid large rooms with short reverb or small rooms with long reverb (which, while highly unrealistic, can create interesting results).
- Damping: Adding softer, less reverberant surfaces to an emulated room. This has the effect of softening the highest frequencies of a reverb signal, allowing a

user to “warm up” a sound that might feel somewhat artificial. The real world equivalent of damping would be to place a rug on a wooden stage, absorbing some of the room’s reverberations.

Performing Music with Playback/Fixed Media

One of the more common types of electronic music the contemporary-classical performer will encounter is music involving playback. This fits squarely into the non-interactive category on the Interactivity Spectrum, and can occupy any position on the effect spectrum. In most cases, playback is executed using a DAW connected to an audio interface, routed to either a mixer or directly to a speaker / set of speakers. Most music for playback is in stereo, but it is not unheard of to encounter a piece that is in mono, quad, or any other variant of speaker count.

If you have the luxury of a live audio engineer, it could be in your best interest to allow them to handle the playback of your track: the fewer wires running across the stage the better. However, it is absolutely possible to perform works with playback without any technical support as long as you have the proper tools.

The most user-friendly method of working with a playback track is to import your files into a DAW like Pro Tools or Ableton. While it is possible to use less powerful digital playback tools like QuickTime or VLC to get the job done, using a DAW will make routing, rehearsing, and balancing MUCH easier. Further, it would be impossible to perform a work with more than two channels in one of these programs.

Locate the audio files sent to you by the composer/publisher, and identify their functions. These may include:

- Performance Track: the track meant to be heard by the audience. This is the only track that is entirely necessary to learn and perform fixed media works.
- Click Track: A track containing a metronome that follows the trajectory of the piece. This track may or may not include elements from the performance track as well.
- Rehearsal Tracks: any number of tracks specifically designed to practice with. This could mean tempo alterations, smaller tracks only covering certain sections of a piece, or some combination of the two. If the work has a click track, it will more than likely be included in the rehearsal track.

Create a session in your DAW of choice, and import the performance and click tracks. The rehearsal tracks are *not* necessary to keep in your performance session, but it can be helpful organizationally to keep them in the same place. If a work doesn't include a click track, proceed to route your performance track to whatever output you are sending to the hall (this will usually be some variant of "1-2", "Main", etc).

There is some extra routing to consider if a work has a click track, so let us establish the outcome we want to reach. The performer needs to listen to the click track and the performance track at the same time, but the audience must only be able to hear the performance track. We can leave our performance track on the main channels, which are presumably being routed to the hall. However, putting the click on this channel will route it to the hall as well, hampering the performance. To this end, we must route the click to its own channel set (possibly "3-4", "monitor", etc), which are being sent to our headphones. The last step in this situation would be to ensure that you as a player can hear the performance track as well as the click. Some composers handle this for you by including the fixed media in the click. If this is not the case, you can either make an exact copy of the performance track and route it to your headphones, or send the main performance track to both your headphones and to the hall (this is possible in most, but not all, DAWs).

In most situations requiring a click, the composer has ensured that the fixed media and clicktrack files are the exact same length, making their alignment fairly straightforward. Simply drag both files to the beginning of the session, and listen back to ensure that your clicktrack and the fixed media are in sync. If this is not the case, you will need to do some searching to find where the click and the performance track align.

The above procedure will see you through most cases where fixed media playback is required, but there are some variables to consider regarding both your hardware and the composer's files. First, hardware: if you are working directly with an audio interface, there is a good chance that there are designated headphone outputs, either ¼" or ⅛". These outputs are sometimes internally routed to listen to all output channels in use, making the need to route the fixed media twice unnecessary.

An uncommon but nevertheless prevalent problem is the media/click stereo split. In this file delivery system, the composer has combined both the performance track and the click into one stereo file, with performance in one ear and the click in the other. The intent here is to allow playback from less robust setups (smartphones, computers without DAWs, etc), and split the stereo file to two locations: the performer's headphones and the audience. To work as intended, the performer must have access to a stereo splitter, a specialized piece of hardware that splits a stereo signal into two mono signals. These are affordable, but if you are working with a DAW you can avoid this step.

Import the track into your DAW as you normally would; it will appear as a stereo file. From here, locate the “split into mono” function and apply it to that track. This will result in the generation of two new mono files, one for each half of the original stereo file. You can now delete the stereo file from your session, and route the mono files as laid out above (performance routed to the hall, click routed to your headphones). Note that because these are mono files generated from a stereo file, it is possible that some strange panning has held over in the transition. Ensure that your performance file is centered, and coming out of both speakers evenly.

We can look to *Timelapse* by Elainie Lillios as a real world example of a work performed with fixed media. The following pair of setup diagrams is included in the performance notes of the score.

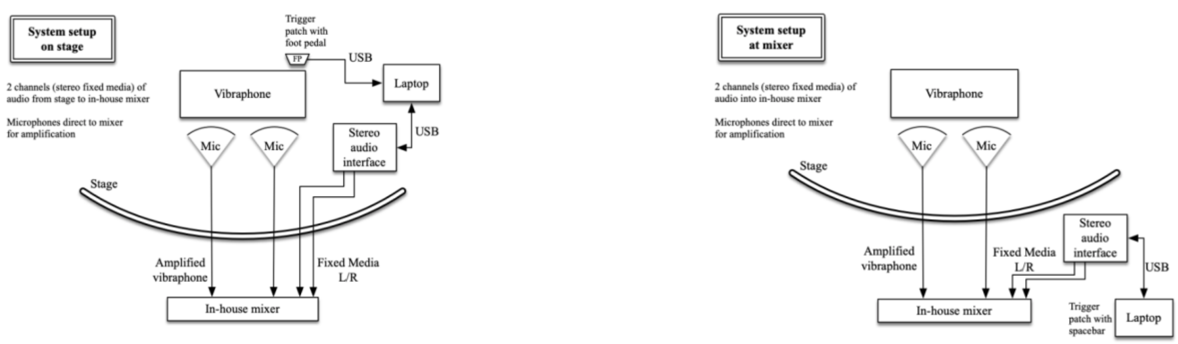

Figure 6. The two suggested routing diagrams included in Elainie Lillios’ Timelapse *from the EIP.*

While these two setups might look intimidating, we can demystify it by simply following the signal. There are two sound sources to consider: the performer and the fixed media track. The track is played back on the laptop, which we can see is connected to an audio interface via USB. The audio interface, which Lillios notes needs to send a stereo signal, is then routed to the mixer (in a less equipped venue, one could also route the interface directly to the speakers). The performer's signal is captured by the microphones and run to the same mixer as the fixed media to be balanced and amplified.

Both of these diagrams are equally viable for performance, and the ideal setup will depend entirely on the performance venue. If a percussionist is playing *Timelapse* in a well equipped concert hall with a built in sound system and an onsite engineer, it might be advisable to provide the engineer with the playback files and work with them to balance the mix and start the track during the performance. If they are playing in a less established space, though, it might be advisable (or even necessary) to play the audio back from the stage. It is also necessary to consider the clicktrack when choosing a setup. If the in house system is chosen, the engineer will have to send the performer their clicktrack in addition to mixing the performance. If the laptop is on stage, the percussionist can simply run their clicktrack out of the headphone output of their interface.

Performing Music with Interactive Electronics

The prospect of performing music with live electronics is one of the more daunting ones in electronic music-making. While some works in this category make use of relatively user-friendly DAWs or pedalboards, it is common for works with live electronics to employ complicated audio programming languages. These languages are often incredibly dense and take a great deal of time to master. For the performer, though, only a cursory understanding of these systems is necessary to get through most works with live electronic processing.

There are a number of incredible resources available for the musician who desires to dive deeper into the world of live electronics and learn more about these tools (see Appendix I). This chapter covers the basic terminology used in live processing settings, and includes the most necessary information to set up and troubleshoot electronics when preparing a performance. Additionally, more detailed explanations are provided for two of the most common languages that a performer might encounter: Max/MSP and SuperCollider.

There are numerous audio programming languages that the performer might encounter when they decide to take on a work with live electronics. Often, the tool chosen for a work depends on the specific expertise of the composer. Each language has strengths and weaknesses though, so a seasoned composer of electronic music might have the luxury of choosing the best tool for the job. Some (but certainly not all) of the languages you might encounter in your career are as follows.

- Max/MSP/Jitter: Max has been referred to as the common language of live electronic music. It is certainly the most common language the performer will encounter, largely due to its flexibility and unique user interface. Max is an object-based visual programming language, meaning individual objects can be connected to each other to create complex systems. It is published by Cycling '74, and licenses are available for subscription or outright purchase. However, it might not be necessary to purchase a license depending on your needs. Max is free to download, but restricts users from saving changes when they lack a license. Thanks to this feature, it is possible to perform works with Max/MSP free of charge.

- SuperCollider: This language is derived from C++. While it is object oriented, SuperCollider lacks the friendly visual interface of Max or PureData. This language is known for having some of the most appealing synthesis in the live audio processing world, making up for its much less intuitive user experience. On opening the program, the user will notice that two iterations of SC appear to be running. This is because SuperCollider is split into two halves, a client and a server. The server (referred to internally as scsynth) is responsible for all audio generation, while the client (sclang) is where the user inputs data. The two halves communicate over Open Sound Control (OSC). SuperCollider is entirely free, and is available for download from GitHub.
- Pure Data: Pure Data (PD) is an object-oriented language with visual representation released by Max developer Miller S. Puckette in 1996. Pure Data is an open-source (and free) alternative to Max/MSP that operates on many of the same principles; while much of the syntax is unique, a user of one will be able to make sense of the other. Because of its open-source nature, PD is less user-friendly than Max, has less robust synthesis, and lacks the exhaustive help documentation that Max has. However, PD is capable of running on Linux machines as well as less powerful computers like Raspberry Pi. Because of this flexibility, PD often gets used by video game developers to generate and test sound design. Video game

developer Electronic Arts has a modified version of Pure Data referred to as EAPd that it uses on projects, like its 2008 release *Spore*.[49]

- CSound: Originally written by MIT engineer Barry Vercoe, CSound is an older language written in C, hence its name. The program has somewhat fallen out of fashion as tools like SuperCollider have come into vogue.

Pedal Logic

Most works with live electronics operate in scenes, which change what processing effects are employed by the computer. There might be a scene with a granular synthesis element, followed by one with heavy reverb, then one with a pitch shifter and delay unit. In an effort to simplify the piece for the performer, some composers use complicated pitch and impact tracking processes to move the piece forward without taxing the player any further. Regardless of the composer's intentions, these methods tend to be somewhat unreliable under all but the best circumstances. The most straightforward way to move through the scenes in a work with live electronics is to employ a food pedal.

If you have received a patch to perform with that requires a foot pedal, it is likely that the composer has included setup instructions. In most situations, following these instructions is enough to get a pedal working. Unfortunately, there is no standard for how a piece takes pedal input largely thanks to a lack of standardization in how foot pedals send information. It is entirely possible that the composer has written their pedal

[49] Kosak, Dave. "The Beat Goes on: Dynamic Music in Spore." GameSpy, February 20, 2008. http://pc.gamespy.com/pc/spore/853810p1.html.

logic for a device different from the one at your disposal. For this reason, a cursory knowledge of how these pedals work can be incredibly helpful.

There are two varieties of triggering pedal that are common to contemporary classical electronic performance: USB and Bluetooth pedals. USB pedals are the most common at time of writing, and often Bluetooth pedals are repurposed page turner devices meant for reading music on tablets. Both accomplish the same thing: they are pressed to send a command to a patch to move it to the next scene. While they are functionally interchangeable except for setup, it is the author's recommendation to make a habit of using USB pedals since they offer a more reliable and direct wired connection, and do not require batteries to operate.

Foot pedals produced by different brands can send a wide variety of information to do their jobs, often dependent on their intended use. Some (most often Bluetooth pedals) even come with different modes that allow the device to interface with more than one type of system. Most often, though, a foot pedal will send ASCII information. ASCII is shorthand for American Standard Code for Information Interchange, and is how computers make sense of the information that we feed them. For example, the ASCII code for the letter "a" is 065. This number translates to the binary number 01000001.[50] More often than not, a foot pedal will send ASCII value 32, which is the code for the spacebar on a keyboard. This means that if a USB pedal is plugged in and pressed, the computer will read it as a spacebar. It is easy to tell if your USB pedal will emulate a

[50] Dodge, Charles, and Thomas A. Jerse. *Computer Music: Synthesis, Composition, and Performance*. Second Editioned. New York, NY: Schirmer Books, 1997.

spacebar: simply start the patch, and press the spacebar. If the patch responds as intended, most USB foot pedals will work out of the box.

In instances where this is not the case, some investigation will be in order to find what the pedal is sending the computer. To accomplish this, most audio coding languages have a function built in to read keyboard input. We will be looking at Max/MSP because of its visual nature, but the logic we use can be applied to any audio coding language.

The process is relatively simple but is not necessarily obvious. First, we must generate an object to take keyboard input, which in Max's case is appropriately called "key". We then generate a number object (hitting "I" on the keyboard will accomplish this), and connect the leftmost outlet of "key" to the leftmost inlet of our number object. From here, we should make sure that our foot pedal is connected and press it. If the pedal is sending ASCII information, the number object will display the ASCII code for whatever key the pedal is emulating. So if 32 appears the pedal is acting as a spacebar, but if it displays a 65 then the pedal is acting as the letter "a". To identify the values your pedal is sending, ASCII tables are readily available for reference online and can be found with a cursory Google search.

Max/MSP/Jitter

Next to fixed media music using a DAW, Max/MSP/Jitter (referred to hereafter as Max) is the most prevalent tool for implementing live interactive musical systems. Named after computer music pioneer and Bell Labs engineer Max Matthew, the software was developed at the Institut de Recherche et Coordination Acoustique/Musique (IRCAM) as a management interface for MIDI.[51] It could not produce or process an audio signal on its own, and was mainly utilized to send highly detailed control information to more sophisticated hardware synthesizers and effect units. After being acquired by software publisher Cycling '74, Max received the MSP expansion (standing for Max Signal Processing, as well as the initials of Max's creator, Miller S. Puckette). MSP allowed Max to process live audio in real time, greatly expanding its versatility as a standalone tool. The Jitter expansion introduced protocols for handling video, and in 2017 Ableton acquired Cycling '74 to introduce Max for Live, a built in interface that allows Max patches to be run in the Ableton Live DAW.

There are a number of excellent resources for learning to write your own Max patches, the first and foremost of which is the built-in documentation. This guide is an introduction to the most vital concepts of handling a Max patch, and will include a discussion on basic terminology, navigation, and tropes that are common in the patches being designed today. While there will be a brief discussion of how to create and edit patches, the primary function of this guide is to help users navigate, not create, in Max. Readers interested in building their own patches should visit the list of resources in Appendix I.

Max is an object-oriented programming language, meaning it contains a number of single purpose modules (or "objects"), that each serve a predetermined function. To work in Max is to string these objects together into more complex functions. Max is unique in that it is a visual coding language: each object is represented by a box on screen, and systems are made by plugging objects into each other. This allows the user

[51] Holmes, 564.

to visualize their signal flow, and makes Max a relatively user-friendly first program for a musician interested in live audio processing.

The Environment

When a user opens Max, the Console is the first thing that appears. This is where Max displays its output data and errors, and can be helpful in ensuring that data is firing when it is supposed to. However, the Console is not the primary way that a user interacts with Max. To get started, we need to open a new patcher by either selecting "New Patcher" from the file menu in the toolbar or by hitting Command -> N.

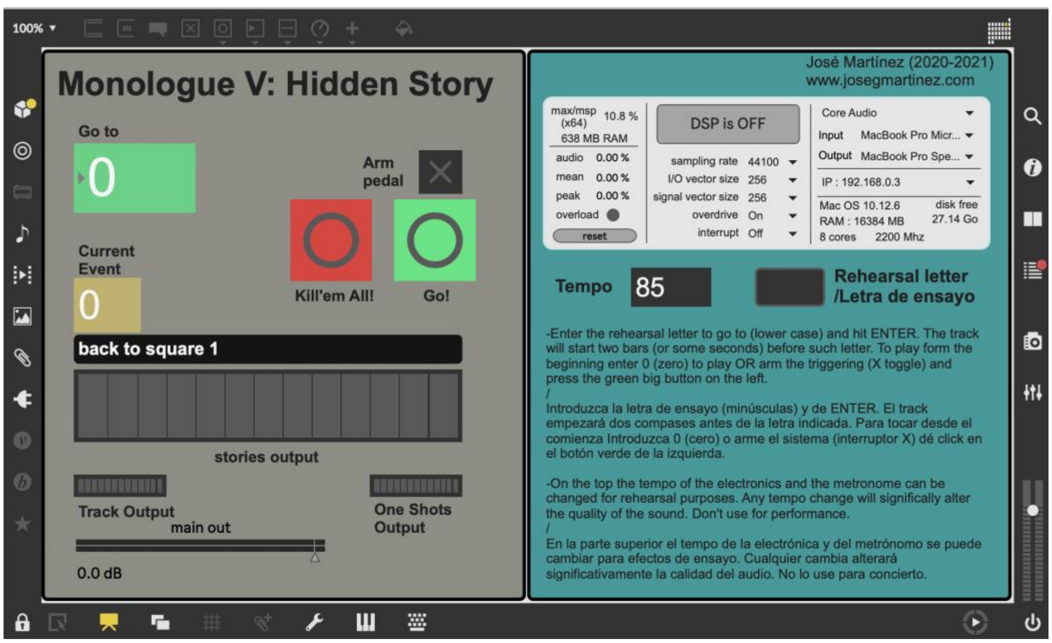

Figure 7. The main patch for Monologue V: Hidden Story *by José Martínez, taken from the EIP.*

Once a blank patcher has been created, a user will find an off-white screen with a grey outline that features a number of menus and functions. While all of these are useful, there are only a few that are necessary for the performer to fully understand.

- Lock/Unlock: Represented by a padlock in the bottom left corner of the window. When locked, editing is disabled and clicking on an object will initiate its function. When unlocked, a user can edit a patch and most object functionality is suspended so they can be connected and positioned correctly. When a patch is unlocked, a user can still activate an object if they desire to by holding the command key and clicking. For example, a bang object cannot be clicked when a patch is unlocked, and clicking and dragging will move the patch around. When the patch is locked, the bang can no longer be moved and clicking on it will activate it. If you are receiving a patch from a composer to perform with, it is highly likely that it will open in locked mode.

- Presentation Mode: Represented by a screen on a stand, and located on the left side of the bottom of the window. Presentation mode in Max is a way for users to hide the messier parts of their code and only show a performer what they need to use the patch. When presentation mode is off, every object in the patch will be displayed. When presentation mode is activated, though, only objects that the composer/use has designated will appear. To change the objects that display in presentation mode, right click on the desired object and select either “Add to Presentation” or “Remove from Presentation” as is appropriate. In Edit mode, objects that will appear in presentation mode will be surrounded by a thin red outline.
- Also note that the positions of objects can be changed between presentation mode and edit mode. When you receive a patch from a composer, it is likely that it will open in presentation mode.

- Audio On/Off: Represented by a power button and located in the bottom right corner of the window. This toggle tells Max whether its audio interface needs to be active or not. While all numerical functions in Max will work when audio is off, *nothing dealing with an audio signal will function until it is toggled on*.
- Console: Represented by a sheet of notebook paper and located on the right side of the window. This will toggle the Max Console display on or off. The console can also be accessed in its own window by selecting it from the Window menu in the toolbar.
- Inspector: Represented by a lowercase i and located on the right side of the window. The inspector is where a user can adjust variables like appearance and numerical range for individual objects. To view an object's inspector, make sure the inspector is activated and select the desired object.
- Objects: Represented by a Max object outline and located on the right side of the window. Contains a comprehensive list of objects available in Max, with different categories to help a user locate what they need.
- Package Manager: Represented by a cube and located on the right side of the window. While Max contains a large variety of objects, there are some third-party objects that do not come with the program. Some of the more common free objects can be located and installed via the package manager. If a composer has used a third party package and not included it with their patch, it might be necessary to download it yourself.

- Object Generators: The group of buttons and menus located at the top of the window. Clicking on these will generate new objects in the Max window. This can also be accomplished by hitting the N key and typing the name of the desired object, or using the shortcuts discussed in the Object Categorization section.

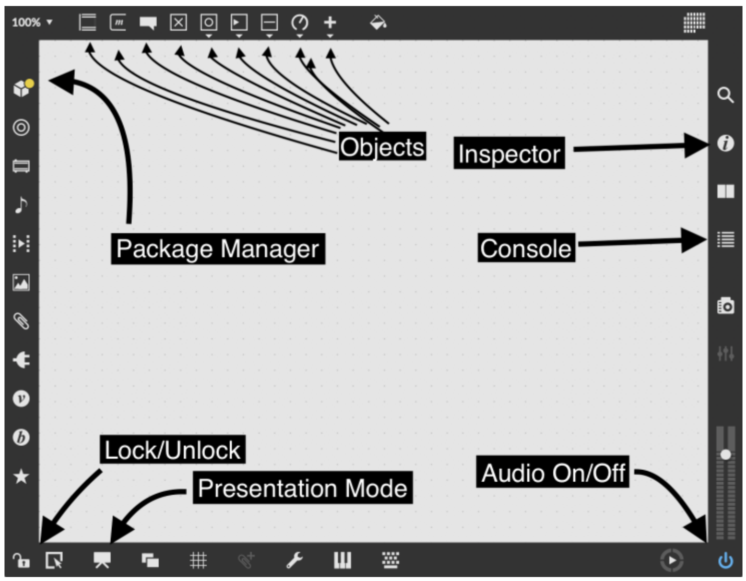

Figure 8. The primary Max/MSP interface.

Functionality

All objects in Max connect to each other over cables using *inlets* to receive data and *outlets* to send data somewhere else. Mousing over an inlet or outlet will give you information about what kind of information the object is looking for, and what it will do with the information provided to it. Further, there are two types of inlets, hot and cold. A hot inlet (indicated by a red outline when moused over) will activate the function of the object and send out a signal. A cold inlet (indicated by a blue outline) will take information in, but will not trigger an output; the information supplied to the inlet is stored until a bang message is received. Which leads into the next question: what is a bang?

Bang messages are perhaps the simplest type of data in Max. Effectively, a bang is an instantaneous message that tells an object to do something. By this logic, a hot inlet is just one that sends a bang along with the input data, and a cold inlet only sends the data. Knowing how objects communicate with each other, we can now discuss the different types of objects and how we can recognize them. There are roughly 690 individual objects built into Max, but a user does not need to know the specific functions of every single one. At a basic level, there are several types of objects that a use should be able to identify.

- Max Object: a rectangular box which outputs message data at the control rate. Can be generated by hitting your “N” key.
- MSP Object: A rectangular box which outputs signal data at the audio rate. The names of these objects are usually followed by a tilde (~). These are also generated by hitting your “N” key.
- Jitter Object: A rectangular box which outputs video data. The names of these objects are often preceded by the qualifier “jit.” These are also generated by hitting your “N” key.
- Message: An object with rounded edges which exclusively stores and sends message data. Can be generated by hitting your “M” key.
- Comment: A clear object that displays text. This object serves no functional purpose, and is used to communicate information to the user or document how a patch works. Can be generated by hitting your “C” key.
- Integer: An object that displays a numerical value. Integers only deal with whole numbers. A user can enter a number into this object via its input,

clicking and typing, or by clicking and dragging the object up or down. Can be generated by hitting your “I” key.

- Flonum: Stands for “floating point number”, float referring to a number with a decimal point. These objects behave identically to Integers, but can deal with a fractional/decimal number. Can be generated by hitting your “F” key.

Max deals primarily with two types of information: *message* data for Max objects and *signal* data for MSP objects. While both kinds of data are often used in a single patch, they serve discrete functions and behave differently. Message data sends in numbers, letters, and signals, and only activates when prompted by user input or a bang from another object. You can tell a connection deals in message data when the cable connecting the two objects is a solid gray color. Message data in Max moves at the control rate, 1000 events per second or 1 event per millisecond. This means that if you string together 3 objects, it will take .0003 seconds for that string of commands to be completed. This is quite slow in terms of computational power, but is fast enough to accomplish the majority of well assembled patches. If we’re going to deal with digital audio though, we need to move information much faster.

MSP objects, signified by the addition of the tilde (~) in their name, are usually used to move a digital audio signal, but can also be used to transmit information much more quickly than Max objects. This is because instead of operating at 1000 events per second, MSP objects move data at a rate equivalent to the sample rate Max is set to, which is usually either 41,000 or 48,000 samples per second (there are other less common options, see page the Digital Audio chapter of this document). This means that

MSP objects can work information fast enough to accurately play back audio files or work with live audio inputs. A user can identify the connection between MSP objects by the checkered green and grey patch cables.

While Max and MSP objects work at different rates, there are some instances where one can take an input from the other. This will generally be indicated by the mouseover text on the input of a given object, where the appropriate types of data will be listed in parentheses (e.g. “signal/float”).

The majority of objects in max display their names and any qualifying data added by the user. There is a small family of highly common objects that include their own graphical information, and are common to many (if not all) patches a performer might run across. These objects give Max its distinct visual look, and know what each one does will help a user to quickly make sense of a patch.

- Ezadc~: Stands for “easy digital audio converter”, and is visualized by a microphone. This object has two states, on and off. When off, Max is not looking for audio data. When on, it looks to your chosen audio device/interface for an incoming signal.
- Ezdac~: Stands for “easy audio digital converter”, and is visualized by a speaker. This object, like ezadc, ezdac has on and off states. When on, ezdac sends audio to your chosen audio device/interface.
- Bang: A bang is the simplest form of data in Max. This object doesn’t have an on or off state, but when clicked or otherwise activated by an input will send out a message. This message is what tells others objects to “do their thing”.

- Meter~: An MSP object that takes an incoming signal and displays it visually. Note that the meter does NOT output audio, but simply displays it.
- Gain~: An object that attenuates an incoming signal and outputs that signal at its new volume. You will find these in most patches dealing with audio in order to control your input/output levels.
- Slider: An object that behaves similarly to Gain~, but deals with message data instead of signal data. These allow users to make adjustments to parameters without dealing directly with the numbers Max takes in.
- Toggle: An object resembling an X with on and off states. They are often used to toggle input and output routes on or off.

Figure 9. From left to right: the ezdac~, ezadc~, bang, toggle, meter~, gain~, and slider objects.

With Ableton's absorption of Cycling '74, a new set of objects was conceived to assist in the crossover between Max/MSP and Ableton Live. These objects, called "Max for Live", are available even when using Max as a standalone platform without integrating Ableton Live, and their names generally begin with "live." (for example, live.gain). These objects can be incredibly useful, but be aware that they have a tendency to require more processing power than their standard counterparts.

Now that we have identified a fairly large (but certainly not comprehensive) variety of Max and MSP objects, we can approach the types of information one will often encounter in the naming schemes of these objects, primarily arguments and attributes. Arguments in Max refer to text that appears after the name of a given object, separated by a space. Most objects can receive several arguments, and the ones available are listed in a drop-down menu that appears when a user inputs a space after an object name. For example, take the cycle~ object, which generates a sine wave. While the frequency of the wave can be received via the input of the object, one can also input the frequency after the name of the object to set it automatically ("cycle~ 440" would generate an A at 440 Hz). Max reads arguments in the order that they appear on the drop down menu. For our cycle~ object, three arguments are displayed along with what kind of data they require: frequency (number), buffer name (symbol), and sample offset (int). If a user attempts to use the wrong kind of data for an argument, their object will turn red to indicate that it isn't working.

The other important object specific data type is the attribute. Attributes are also listed on the drop down where arguments are defined, and are methods of telling an object how to behave. To set an attribute in an object name, insert the "@" symbol followed by the attribute you want to set, then input the data after the attribute name. Attributes can also be fed to an object via a message box, or directly altered in the inspector for an object.

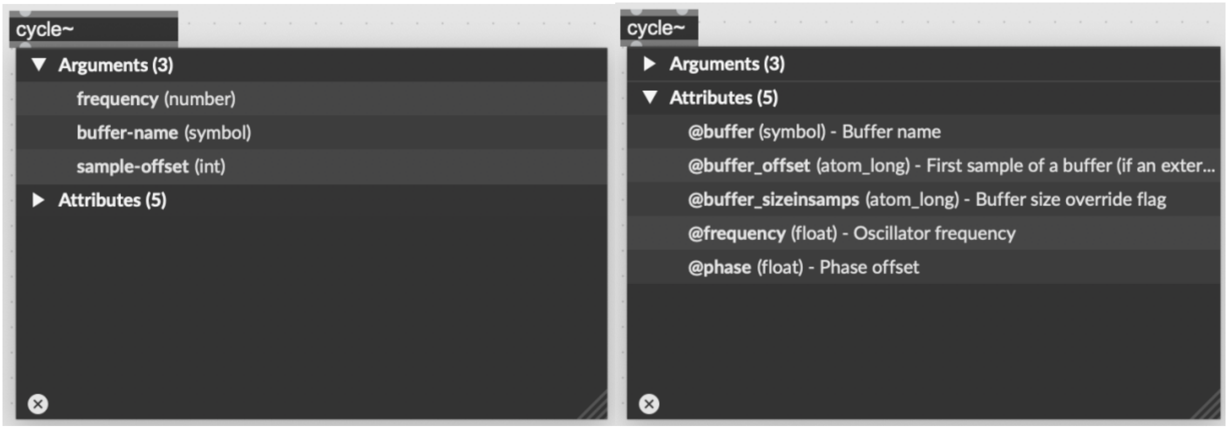

Figure 10. The arguments (left) and attributes (right) available for the cycle~ object.

Given the vast variety of objects available to a Max user and the large number of objects that it takes to perform even the most simple tasks, it can quickly become difficult for users to keep their patches clean and easy to use. Additionally, it can be incredibly labor-intensive for a user to if a patch requires the same organization of objects in several places. For this reason, the savvy coder will often make use of subpatchers and abstractions.

Subpatchers (indicated in Max as “patcher” or simply “p” followed by a custom name decided by the user) can be thought of as a patch within a patch. Double-clicking on one of these objects will open a new window that operates the same as any normal patch; they can contain any combination of objects connected in any configuration. They will normally contain inlets and outlets which allow a path for information in and out of the subpatcher.

Subpatchers are useful organizational tools, allowing complex code to be condensed down into a single object. In particularly robust patches, it is not uncommon to find subpatches within subpatches containing repetitive or messy code. A defining factor of the subpatch is that they exist *only* within the given patch, meaning that they are not saved anywhere except for the patch file they are written in. If a user wants to perform a similar function but save it as its own file, they can employ abstractions.

Abstractions are, functionally speaking, nothing more than subpatchers saved as their own file. In this way, any file can be loaded as an abstraction by simply generating a new object and typing the name of another Max file. The appeal of the abstraction is that it allows a user to create a patch that fulfills a function once and use it in any of their other work where it may be of use. An abstraction must be saved in the same folder as the file it is being used in, which is why which is why when a performer downloads a new piece with Max they will often find a folder containing many files (It is possible to tell Max to look for an abstraction in a different location on a hard drive, but most performance-based users will not need to make use of this function). In these situations, most of the extra files are abstractions referenced by the main patch, usually titled something like “MAIN” or “OPEN_ME”.

When an abstraction is written it will more than likely need inlets and outlets just like a subpatcher, with a minor aesthetic difference. Rather than displaying a number to indicate how many inlets / outlets there are, they will simply display an “I” for inlet or “O” for outlet. This indicates that the ins and outs are at the top level of the patch. To complicate things further, it is entirely possible for a user to write an abstraction that references other abstractions or even contains subpatchers.

With all of this data floating around, it is understandable that a first-time user might find Max intimidating. The good news is that in most patches, most of this information will not concern the performer, and most of it will more than likely be hidden from view. Max allows users to create a specific visual version of their patch in what is called *presentation mode*, where the only objects that a performer might need to interact with are visible. Presentation mode can be toggled on and off by clicking the small projector screen button on the bottom left of the Max window, but if you are performing a patch designed by a composer it is likely that they delivered the patch to you with presentation mode already activated.

Objects do not automatically load into presentation mode, and need to be added manually by the user. To accomplish this, right click on the object you would like to add, and click “Add to Presentation”. Objects can be removed from presentation mode in the same manner. When the patch is unlocked, objects that are selected for presentation mode will be outlined with a dull pink border. Note that when presentation mode is active, objects not selected to display are still there and are still functioning: they are just hidden from view to keep the patch from getting cluttered (or to keep a performer from changing vital parts of the code).

When presentation mode is active, Max’s GUI stops displaying certain items (like patch cords) and allows the user to move objects wherever they wish. This allows users to design their patch with logic in mind, then condense it down so that it is more user friendly. If presentation mode is deactivated, it is possible that objects will suddenly shift positions rather drastically. Don’t panic: they have simply reverted to their original placements, and reactivating presentation mode will return them to their familiar locations.[52]

[52] Cycling '74. “Max/MSP/Jitter.” Program documentation. *Https://Docs.cycling74.Com/max8*. Cycling '74, September 25, 2018. https://docs.cycling74.com/max8.

SuperCollider

While Max is by far the most common tool for making interactive electronic music, it is not without its shortcomings. The user-friendly visual interface can be a hindrance in writing efficient code, and many composers find the quality of its digital signal processing less than desirable. Users who prefer a more streamlined and text-based interface often turn to SuperCollider.

SuperCollider is an audio programming language rooted in C++ designed for algorithmic composition and digital audio synthesis. It was developed by James McCartney in 1996, and was rereleased as free software in 2002. SuperCollider is known for its efficiency, but can be frustrating for users new to electronic music. Anecdotally, I have found that users with some background in computer science often prefer SuperCollider, while those without any background find it more frustrating than the likes of Max or Pure Data.

There are two primary parts of SuperCollider that a user needs to be aware of: the server (SCSynth) and client (SCLang); this is why it appears that two programs are running at once when SuperCollider is running. The server is primarily focused on audio production, and features the basic building blocks of SuperCollider, called UGens. The client both sends and interprets commands for the server via the Open Sound Control (OSC) protocol. By separating the two functions, Supercollider is safer to perform with (if the client crashes, the server will continue to generate sound), and can be controlled by both external programs and offsite computers.

The Environment

There are three primary parts of the default SuperCollider window.

- Text Editor: Where a user writes and runs code, located on the left of the default window. This is where you will spend the most time in SuperCollider, and what is loaded when you load a SuperCollider save file.
- Help Browser: The documentation for how SuperCollider works, located in the top right of the default window. This is where you can read about the inner workings of SuperCollider, work through tutorials, and find definitions for the components that are entered in the Text Editor.
- Post Window: Where SuperCollider outputs data, located in the bottom right of the default window. This is where SuperCollider displays both it’s successful output and any errors that may occur.

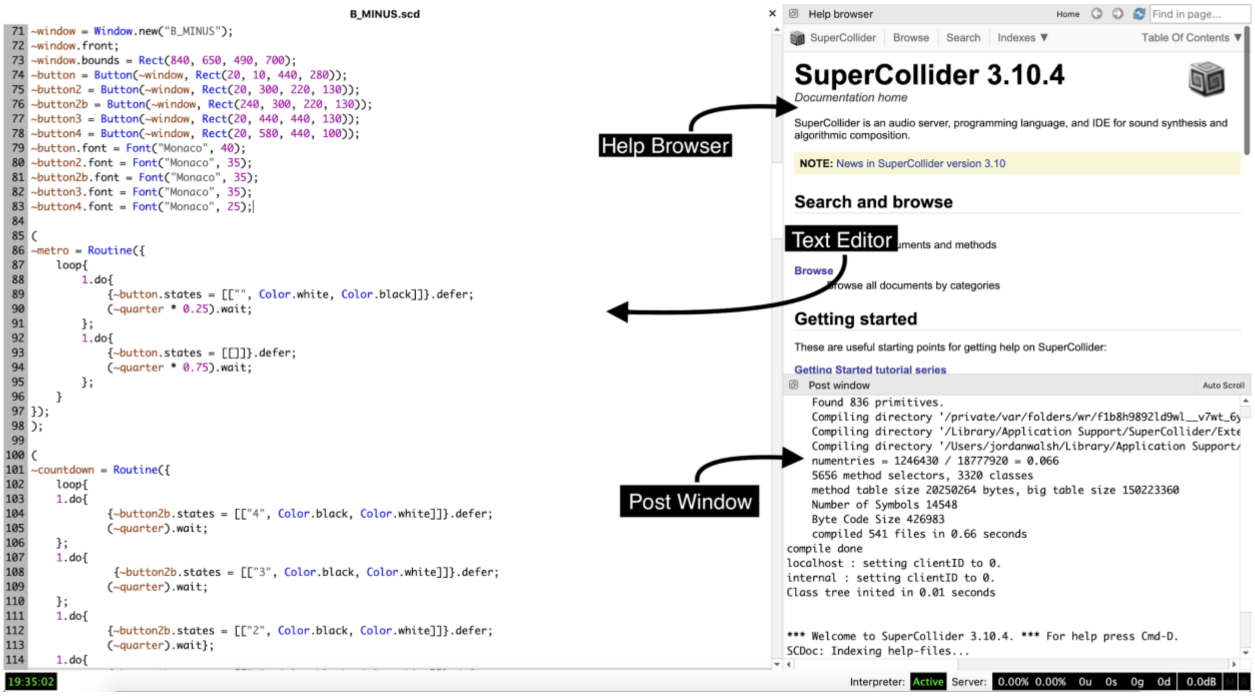

Figure 11. The primary SuperCollider interface.

Additionally, SuperCollider displays a status bar at the bottom of the window to monitor the state of the Interpreter and Server. The text here is color-coded green (everything is functioning), white (inactive), and yellow (something isn't working).

Functionality

As a text-based language, SuperCollider lacks an easy way to create a graphical user interface for a patch. This means that when a composer delivers a piece with SuperCollider the user will be shown the code as opposed to the more friendly "presentation mode" like in Max (some composers will go to the effort of creating a user interface, but SuperCollider will not display it until the patch is run). For this reason, it is fairly necessary to have at least a cursory understanding of how SuperCollider handles information in order to perform with it.

SuperCollider thinks in lines of code that are separated by semicolons. The to execute a line of code, place your cursor on a line and press command -> Enter. To stop a line of code that is already running, press command -> period (.).[53] To activate several lines of code at once, a user can place parentheses at the beginning and end of whatever they wish to execute. In relatively simple patches, there might be parentheses around all code in the text editor. In these situations, evaluating the file (Command -> Enter) will run the piece.

There are a few symbols that tell SuperCollider how to interpret input in the text editor.

---

[53] Both of these functions can also be accomplished by finding them under the “Language” Menu in the toolbar.

Symbol	Name	Meaning
//	Comment	Any text on a line following this symbol will be ignored by SC. This is useful for making notes of how code works or leaving instructions for performers.
/* */	Expanded Comment	Performs the same function as //, but accounts for multiple lines of text. Any text before /* but after */ is considered a comment by SC.
;	Line Separator	While SuperCollider thinks in lines in some cases it needs to be definitively told that a line has ended. For this reason, it is good practice to place a semicolon at the end of each line of code.
()	Parentheses	Used to group lines of code together. When a user executes a line within parentheses, all lines in that group will be executed together.
{}	Function	A container for a mathematical function. Most useful when defining a formula that will be used repeatedly, or handling data that changes over time like live audio.
[]	List/Array	A collection of data that can be recalled together, or altered. Items in a list are separated by commas.
“ ”	String	Contains characters for display, and is often used to name synths or display text in a GUI. Evaluating a string on its own will display the contained text in the Post Window.
\	Argument	One of the more flexible (and therefore difficult) symbols. For our purposes, its primary function is to access and argument within a Class.

Table 3. Common types of syntax used in SuperCollider

Each of these symbols changes the way that SuperCollider thinks about your input. For example, say we placed the number four (4) in a string and in a list. Since strings only display data, they cannot be used to perform any math. In a list, though, SuperCollider sees the four not just as a character, but as the number four. If that list gets referenced in the program, it will retain its numerical value.

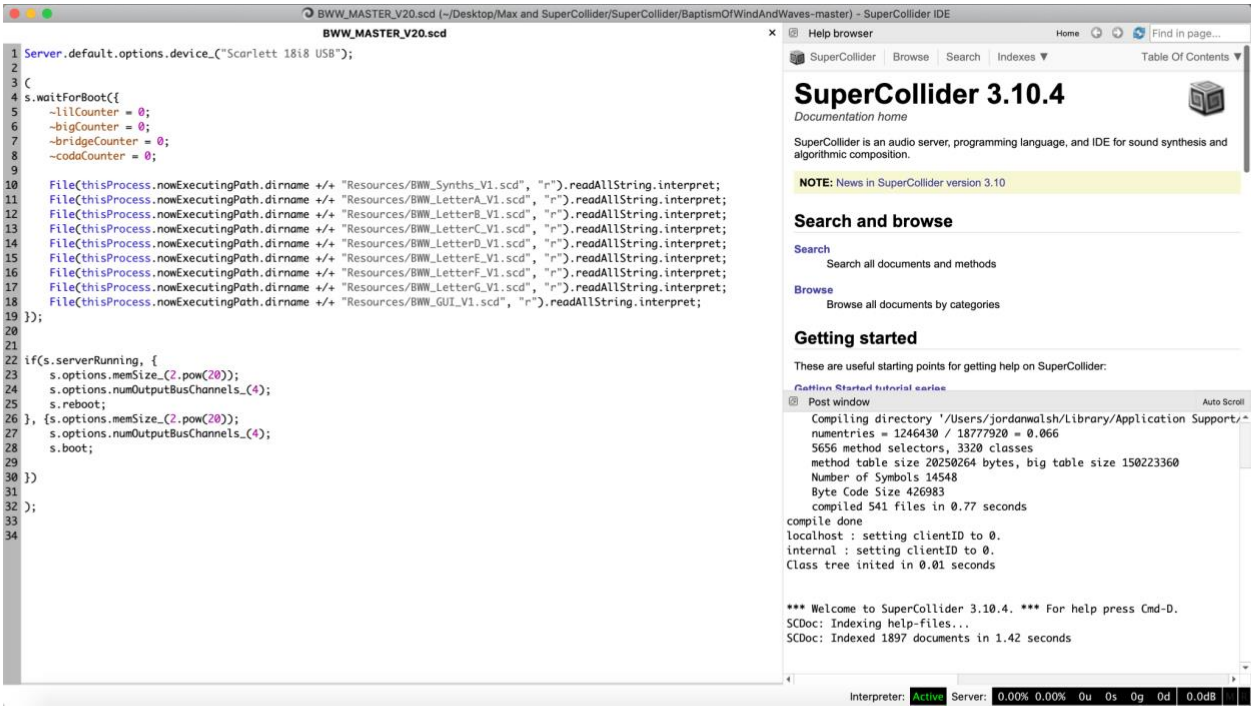

Figure x. The main patch for James Parker's "Baptism of Wind and Waves" from the EIP.

While these symbols tell SuperCollider how we intend to use data, it still does not know what we are doing with it. For that, there are several other types of information to be aware of.

Information	Use	Syntax Example
Class	In SuperCollider, a class is something that performs a job. For example, the “SinOsc” class generates a sine wave. Classes always start with an uppercase letter, and contain methods and arguments that a user can alter to various outcomes.	SinOsc
Method	Methods tell classes to do something, and usually follow the class and a period (the method “ar” on the class “SinOsc” would read “SinOsc.ar”.	SinOsc.ar
Argument	Data contained inside of parentheses in classes and functions that fill certain roles. In our “SinOsc” class, there are arguments for frequency, phase, mul, and add, defining these variables for when the class is activated. When we open up the parentheses of a class, a small menu will appear to remind the user what arguments are available. While class arguments are sensitive to the order in which they appear, users can skip to specific arguments by typing the argument’s name followed by a colon.	SinOsc.ar(440, 0, 1, 0); SinOsc.ar(phase: 0, add: 1);
Local Variable	Variables that are defined within a certain function, class, et cetera. SuperCollider only knows about these variables within the function where it is defined. If a user tries to call a local variable outside of its home, SC will throw an error.	(Var thing = 10;)
Global Variable	Variables that are usable anywhere in a SuperCollider program. SC sets aside the letters of the alphabet (a - z) as global variables, but it is not recommended to use them since it is easy for them to get redefined in more complicated patches. The preferred method of naming a global variable is to put a tilde in front of a unique word.	x = 10; ~ExampleofGlobalVariable = 10;

Table 4. The most common types of data a user will run into in SuperCollider.

Two of the most common methods that a user will encounter are .ar and .kr, which refer to audio rate and control rate respectively. If you've read the Max chapter, remember that most object based audio programming languages operate at two distinct speeds in order to work more efficiently. If a class is generating, altering, or passing an audio signal, the .ar method is necessary to tell SC to use the audio rate to interpret the class. If the class is not handling audio and is, for example, simply generating numerical data to be used by another class, a user can use the .kr method, which is slower and therefore less taxing on the server.

Because all audio is generated by the server side of SuperCollider (SCSynth), it is often necessary to boot the server before running a patch. This will be indicated by the text on the "Server" portion of the status bar being green as opposed to white or yellow. When you start SuperCollider, the server is defaulted to off. There are a few ways to remedy this.

- Select "Boot Server" under the Server menu on the toolbar.
- Press Command -> B on your keyboard.
- Evaluate a line of text in the Text Editor that reads "s.boot"

Some composers will either write instructions to help you through this process in the comments of their code, or might even write "s.boot" into their piece. Be sure to take a glance at the Text Editor before you begin to see what they might have contributed.

Now that the server is booted, a user should ensure that their input and output match what they want to use for their performance. By default, SuperCollider identifies a default input and output when it is booted by checking the settings of the computer. So if a computer’s input is set to “Built-In Microphone” and the output to “Built-In Speakers” when the server is booted, these devices will be used by SC. Alternatively, a user can enter and execute the following syntax to change ins and outs after a server has been booted.

```
Server.default.options.inDevice_("INPUT DEVICE NAME");
Server.default.options.outDevice_("OUTPUT DEVICE NAME");
```

If you are unsure of what devices are available, a list can be generated by executing this syntax.

```
ServerOptions.devices;
```

When booting the server, the most common error is a mismatched sample rate. This occurs when the active input and output devices are operating at different rates (for example, an input set at 44,100 and an output at 48,000). To remedy this, a user should enter their computer’s audio settings and ensure that their input and output match rates. On a Mac OS X computer, this can be located in the Audio MIDI Setup program.

Now that we have booted the server, look for instructions on how to begin the piece located either in the score or in the comments of the patch. If they are present, starting the piece is as simple as following these instructions. If not, a little more investigation and trial-and-error might be necessary.

Like in Max, SuperCollider users often use abstractions to keep their programs organized and to reuse multifunctional code If a folder delivered by a composer contains more than one .scd file, this is more than likely the case, and you should make sure that you’ve opened the main file. Just like in Max, composers often name their main files things like “Main”, “Master”, and “Open Me”. They may also use capitalization to catch a performer’s eye, or highlight the file to indicate its importance.

It’s About Outcomes

This section has covered a wide range of electronic musical tools and how they work. These explanations have been purposefully presented in an unorthodox sequence in an effort to introduce concepts in relation to how a contemporary classical performer is most likely to encounter them. By linking these fundamental concepts with their real world applications, we have organized an electronic musical pedagogy that is expressly catered to performers, and provides external support for readers who might want to engage with these concepts further. As stated at the outset, this document is not intended to serve as a comprehensive explanation of contemporary classical electronic music. There are a myriad of topics like synthesis, external MIDI controllers, history, and less common coding languages that have either been addressed in passing or left out altogether. The topics presented here are those that I find vital to engaging with electronic music as a performer.

Engaging with electronic music making tools as a contemporary classical performer often feels like an insurmountable task because of the wildly different vocabulary of the practice compared to that of instrumental performance. For performers finding themselves in this position, I find it helpful to remember that electronic music making is an outcomes oriented activity. While there are better and worse ways to use these tools, a successful user is one that reaches a desired outcome, no matter the methodology. Much like a traditional musical instrument, the best way to get to know electronic media is to experiment and practice.

Conclusions

In this document, we have covered a considerable volume of information to support the performance of contemporary classical music with electronic media from both theoretical and practical perspectives. By developing a foundational conception of the works of Pierre Schaeffer and Edgard Varese, we are able to reframe the peculiarities of electronic music and understand how the medium functions differently from acoustic music. We then used this foundation to develop new terminology to better understand the functional nature of electronic music with live performers. The term *audio-experiential discordance* puts a name to the jarring experience of hearing a signal that differs from our assumptions about its source, and the *Interactivity & Effect Spectra* are useful in conceptualizing the role of the performer in this music.

Practically, we have explored the most common electronic music tools from the perspective of the performer by inverting traditional pedagogy. By teaching these tools from the top down, we are able to tackle concepts in the order and to the depth that performers encounter them in the real world. While this method does not lead to a comprehensive knowledge of any of these tools, it can lead to a functional understanding that breaks down the barrier to entry of contemporary classical electronic music.

While we have covered a fairly large variety of topics, there is still much work to be done in reimagining electronic musical pedagogy for contemporary classical performers. Future additions to this project will include a more robust exploration of historical compositional and listening practices to further inform our theory and more comprehensive practical explanations to aid performers in understanding topics not covered here. Further, I plan to continue to commission works for the EIP in order to both contribute to the literature and to use as teaching pieces for students interested in electronic music.

Appendix I: Further Reading

Modern Recording Techniques

David Huber & Robert E. Runstein

Huber and Runstein's *Modern Recording Techniques* is written with the aspiring audio engineer in mind, covering concepts like studio setup, microphone selection, and the science of hearing. The audience of this resource is largely musicians in the popular music recording industry, meaning that some issues that relate specifically to contemporary classical performers (interactive electronics, classical microphone placements) go unmentioned. That said, *Modern Recording Techniques* offers an introductory explanation of all basic recording technology concepts, and is of use to any musician who anticipates spending time in the studio.

Classical Recording: A Practical Guide in the Decca Tradition

Caroline Haigh, John Dunkerley, & Mark Rogers

Compared to *Modern Recording Techniques*, *Classical Recording* is a much more granular look at recording technology practices as they relate to the classical tradition. The book is split into three sections, covering practice in preparation for, during, and following recording sessions with classical music ranging from the soloist to the full orchestra. While much of this information is meant for recording engineers working with classical organizations, there are subsections on recording practices for every common instrument, which is vital knowledge for the modern performer.

The Computer Music Tutorial

Curtis Roads

This work perhaps is the most comprehensive guide to audio technology and computer assisted music making in the history of the field. It contains exhaustively detailed descriptions of synthesis techniques, digital audio, signal processing, MIDI technology, psychoacoustics, and more. While *The Computer Music Tutorial* is a largely technical document not for the faint of heart, Roads has exhaustively referenced his sources and backed up his data with helpful visuals. It is not the friendliest source on audio technology, but it contains information on everything an aspiring electronic musician might need to know.

Computer Music: Synthesis, Composition, and Performance

Charles Dodge & Thomas A. Jerse

This book provides a middle ground between the accessibility of this document and the depth of tomes like Roads' *The Computer Music Tutorial*. Last printed in 1997, *Computer Music* has aged well because it focuses on timeless concepts instead of platform specific information. This is an excellent place to start for the musician who has experience with digital audio but lacks any formal training.

Electronic and Experimental Music: Technology, Music, and Culture

Thom Holmes

Holmes' *Electronic Experimental Music* is a well organized broad look at all aspects of experimental electronic music making. Among other things it includes basic audio technology information, a history of tape music, and a snapshot of some of the most important products and instruments that have shaped the experimental music scene of today. The sixth edition of this text has modularized its structure, making it an excellent quick reference.

Pink Noises

Tara Rogers

Tara Rogers *Pink Noises* is a collection of interviews conducted by the author with twenty-four women who have helped to shape the electronic music making field. As in other musical niches, electronic music history has largely been written around male practitioners and left women out of the conversation. Roger's work fills in some of these gaps, offering a necessary feminist viewpoint to a too often often male conversation.

Max/MSP/Jitter for Music

V.J. Manzo

Max/MSP/Jitter for Music is a highly accessible beginner's guide to all three major components of Max. This book is almost entirely focused on technical execution and its examples are appropriately straightforward. While excellent for beginning students interested in Max, this approach does not display the full creative potential of Max as a tool.

The SuperCollider Book

Scott Wilson, David Cottle, and Nick Collins

The SuperCollider Book is a tome that is greater than the sum of its parts. Each chapter is contributed by a different author, each offering a unique area of expertise within the language. The editors structured the resource to accommodate both total beginners and seasoned users. Section 1, titled *Tutorials*, offers a more or less linear fresh approach to learning the language, while the rest of the book contains more open ended, topics based discussions on specific applications of the language.

Appendix II: Documentation

Max/MSP

Examples: Max comes preloaded with a variety of examples for how to build common configurations and effects, which can be helpful when a user needs to build something fairly standard (like a reverb unit) quickly. Max examples are houses in the Help menu on the toolbar.

Help Files: Each Object in Max/MSP has a designated help file that explains how it works. To access an object's help file, right click on the object and select "open [object name] help" at the top of the menu, or Option -> Click on the object.

Reference: A much more in depth explanation of each object and concept in Max. To access the top level of the Max Reference, select "Reference" in the Help menu on the toolbar. To access Reference files for specific objects, right click on an object and select "open [object name] reference".

SuperCollider

Help Browser: By default, SuperCollider's help files are prominently displayed in the top right of the window, and contain easy access to all of the documentation, tutorials, and a glossary of terms. To quickly locate the help for a specific object or class, highlight the object in question and press Command -> D.

Appendix III: Access

Max/MSP

Max is available for download from https://cycling74.com/. Once installed, a user can run any patches they receive from a composer, but cannot make and save changes to them without a paid license. If you want to write your own patches, licenses are available for monthly/annual subscriptions or one time purchases. Education discounts are also available.

As Max receives updates, functions within the language are subject to change, meaning that some older works might not function properly in the most recent version. To remedy this, Cycling '74 has made older versions of Max available here under "Other Downloads".

SuperCollider

SuperCollider is a free program under the GNU (General Public License), and is available at https://supercollider.github.io/. Because SC is open source, the developers often make test releases available to the public for the technically minded. For strict performance purposes, be sure to download the "current version", which has been confirmed to be stable.

Reaper

Reaper is one of the most affordable DAW's available on today's market. While a commercial license costs a fairly standard rate of $225, individual users and educational institutions are eligible for a permanent license for $60.

Ableton Live

There are several versions of Ableton Live available for purchase, leveled "Intro", "Standard" and "Suite". Each progressive level includes additional capabilities, the most notable of which are additional synthesizers/sounds and Max/MSP integration at the "Suite" level. Student/educator pricing is available, and no interest payment plans were recently introduced: this is strongly recommended for musicians who anticipate needing the robust capabilities of the "Suite" edition but are on a budget.

Pro Tools

Avid's Pro Tools might be one of the less friendly experiences available today, but is still recommended to users primarily interested in recording. Avid offers both subscription based and perpetual licenses as well as significant educational discounts. Pro Tools employs iLok, a physical licensing system used to evade piracy by locking a user's license to a special USB stick that has to be plugged in for use. While options exist to get around iLok, they are (in my personal experience) not terribly reliable.

Logic Pro

Apple's Logic Pro represents one of the most affordable options for professional digital audio workstations, coming in at a standard price of $200. Even better, Apple offers a student bundle including Logic, Final Cut (video editing), Mainstage (live music performance), Motion (3D animation) and Compressor (video compression) for the same price.

Audacity

Possibly one of the most accessible tools for recording technology is Audacity, an open source DAW. Audacity is totally functional and is absolutely recommended for learning how audio technology tools work. It is also plenty capable of simple routing capabilities for performance of a work with playback.

Appendix IV:

The Electronic Integration Project

The Electronic Integration Project (EIP) is a commission-centric companion to this document which aims to demystify electronic music making practices by practical application. I collaborated with seven composers between 2018 and 2021 to bring seven new works to life that exemplify the potential of electronic media in contemporary musical performance. I have categorized each work on the interactivity and effect spectra in an effort to apply the terminology we defined in Part I of this document. Recordings of all EIP works are available at jordanwalshmusic.com.

My Battery is Low and It is Getting Dark - Brian Ellis

Interactivity Spectrum: Degree 3/4
Effect Spectrum: Degree 2/3

Brian Ellis' contribution to the Electronic Integration Project is an elegy for the Mars Opportunity Rover, which sent its last transmission on October 10, 2018. The title for the work is derived from a poetic English translation of the rover's final message. Ellis describes *Battery* as follows:

> "The form of the piece is the story of Opportunity's journey told backwards. It begins with a feeble string of information passed quietly back to the audience - the "my battery is low" message. As the piece progresses, we go further back in time as more tools, sensors, research equipment is brought online, and more musical elements are introduced… The work ends with a giant crescendo… before finishing with a dramatic upwards glissando across the range of the instrument, the exact reverse of the dramatic landing Opportunity made in 2004."

Technically, *Battery* is a work of generative music. The performer is given a Max/MSP patch and is directed to Ellis' website (brianellissound.com), which generates a new score and a cue list that accompanies it for each performance of the piece. A performance of *Battery* consists of the performer reading through the score, improvising on each cell as they go. While the performer plays, they are also using a foot pedal to tell Max to record each cell as they go. The patch then fades the recording into the mix. Depending on the contents of the cue list, the recording might be pitch-shifted or otherwise modified.

The self-composing nature of *Battery* directly reflects Ellis' thematic content. The public discourse surrounding the end of Opportunity's mission was one of mourning, especially after the phrase "my battery is low and it is getting dark" was coined. The phrase lent the rover a sense of life, and its slow loss of battery life before being declared effectively dead felt tragic. Ellis' work reflects this artificial feeling of life by imitation. *Battery* records, loops, and alters the performer's sound over time, adding systems on top of systems until it feels like the machine has taken on a life of its own.

Battery sits between degrees 3 and 4 on the interactivity spectrum. It is semi algorithmic in the sense that it's score is freshly generated by the computer for each performance. However, because the computer is not actively composing during the performance, it does not quite fit the description of algorithmic music. Regarding the effect spectrum, *Battery* sits between degrees 2 and 3 depending on the iteration of the patch. When the work begins, it is clear that the electronic media is a modified recording of the performer's actions (degree 2). As it progresses and the texture gets thicker, the electronic sounds slowly develop their own sonic aesthetic separate from that of the performer (degree 3).

While the sonic aesthetic of Ellis' piece could theoretically be achieved with fixed media (or even living performers), the thematic content would not carry over this way. Because it is generative, *Battery* will never be performed with the same score in the same way twice. It feels alive in the same way that Opportunity felt alive. Both exist once, communicate something to us, and disappear into our memory. Ellis' *Battery* is a statement on mortality, and could only work as a piece of generative media.

Conversation - Caleb Evans

Interactivity Spectrum: Degree 1
Effect Spectrum: Degree 2

Conversation by Caleb Evans is one of the more performatively straightforward works in the EIP. Its electronic media are strictly fixed media, and the performer plays along with a clicktrack. The piece takes its cues from electronic dance music and film scoring traditions, employing bombastic bass drops and twinkling keyboards. The vocal content of the work is derived from the introduction of the classic radio serial *Suspense*, pitch-shifted down and accompanied by synthesizers for dramatic effect.

The simple nature of *Conversation*'s electronics makes it fairly easy to categorize. It clearly occupies degree 1 of the interactivity spectrum since there is no variable dialogue between the track and the player. Concerning the effect spectrum, the piece employs degree 2 with its vocoded vocal samples and degree 4 with its synthesized piano and bass sounds.

Time-Lapse - Elainie Lillios

Interactivity Spectrum: Degree 1
Effect Spectrum: Degree 1

Elainie Lillios' *Time-Lapse* is a work for vibraphone and fixed media that fits quite precisely into degree 1 of both the interactivity and effect spectrums: the fixed media nature of the piece means that the performer is beholden to the track, and there is no alteration to the sound of the vibraphone. Lillios' fixed media primarily features synthesis, but there are also snippets of vocal samples present. *Time Lapse* represents something of a departure from Lillios' usual output, taking heavy influence from the more pop oriented EDM culture. This results in an interesting fusion between her noisier contemporary style and the more rhythmically propulsive music of modern popular music.

Monologue V - Jose Martinez

Interactivity Spectrum: Degree 4
Effect Spectrum: Degree 2

Monologue V is written for bass drum and live electronics, and contains some of the most direct interplay between electronics and theme in this collection. Martinez's work explores the difficult cultural inheritance of mixed race people, who often struggle to identify with their heritage. To explore this concept, Martinez has developed an algorithm to compose new sampled loops in real time during performance, and requires the performer to listen and mimic the resultant patterns. In this way, Martinez is mirroring the performative nature of engaging with heritage as a mixed-race person. There is a degree of invention in play when learning something on the fly; the performer will more than likely get small details wrong, and might even create news ideas to fill in the spaces that they don't quite understand.

Monologue V is a unique variant on the interactivity spectrum. There is no instrumental processing, partially due to the fairly narrow capabilities of the bass drum as a sound-generating force. Instead, Martinez includes a set of predetermined samples with the patch which are ordered and activated in new ways with each performance, making up the patterns that the performer copies. While the static nature of the samples might first lead a performer to categorize it as degree 1 on the interactivity spectrum, the piece is better suited for degree 4 because these samples are being used as compositional material by the patch.

Baptism of Wind and Waves - James Parker

Interactivity Spectrum: Degree 3
Effect Spectrum: Degree 1

James Parker's *Baptism of Wind and Waves* is the lone work in the EIP written in SuperCollider, and is performed on glockenspiel. The piece operates on a fixed trajectory, and features a pitch-tracking external that activates a choir-like synthesizer and white noise generator. This marks the only work in this collection that features live electronics and does not require a foot pedal to perform.

Baptism fits quite snugly into the 3rd degree of interactivity; the patch is actively listening for specific input from the musician to perform its tasks in real time. During certain parts of the work, the glockenspiel is granulated and played back while the performer improvises, creating a blur between what is real and what is electronic. Concerning effect, the piece partly occupies degree 1, as it is not making any notable alterations to the sound of the glockenspiel (the granulation is light enough that the instrument's tone is left intact). It also evades the scale with its heavy use of synthesis.

The computer acts not as a filter of the musician's work, but as a second performing force of equal or greater importance to that of the glockenspiel.

Inquietude - Jonathan Andrew Smith

Interactivity Spectrum: Degree 2/3
Effect Spectrum: Degree 2/3

Jonathan Andrew Smith's *Inquietude* is written for kalimba and live electronics in Max/MSP. The instrument in question is the specific tuning configuration required for the third movement of Per Norgard's *I Ching*, and requires a kalimba with a pickup to function. Smith's work makes use of a wide array of audio effects including delay, pitch shifting, and convolution, making it an excellent example of how signal processing can create a great variety of sounds from a relatively simple source.

Inquietude fits somewhere between the second and third degrees of both the interactivity and effect spectrums. Concerning effect, the piece features places where simple delays are used to create a somewhat stochastic aesthetic, but in other places pitch shifts the instrument in such a way that it is no longer immediately recognizable. These sections come further into the piece though, so an audience has had time to familiarize itself with the instrument's sound and will more than likely infer that the more drastically modified textures are still generated by the kalimba. Concerning interactivity, *Inquietude* employs both simple responsive alterations (like our previously mentioned delays) that fit into degree 2 and more dynamic effects like granulation with ever shifting parameters that are more at home in degree 3. What makes Smith's work truly interactive is that while the performer has no control over direct control over the parameters of the electronics, they are encouraged to improvise to achieve their desired result. The interactivity is between a semi-fluid system and a performer that has to operate within it.

Particle Wave - Kirsten Volness

Interactivity Spectrum: Degree 2/1
Effect Spectrum: Degree 2

Kirsten Volness' *Particle Wave* is a work for solo vibraphone with Max/MSP, and is one of the best examples in this collection of how a work might occupy multiple levels of our interactivity and effect spectrums. The electronic elements of the work are fairly simple, with digital delay and reverb being the two primary ways that Max alters the vibraphone's sound, positioning the piece firmly in the 2nd degree of both spectra. However, this paradigm is challenged during the second movement. The strict traditional notation of the first movement falls away in favor of a more time-based aleatoric style of writing, and a fixed media track is introduced. By introducing fixed media halfway into a piece that has thus far primarily featured live signal processing, Volness flips the role of the musician mid-performance. Movement one is fairly performer centric; the audience understands that all sounds being generated originate from the vibraphone. Once the fixed media is introduced, this dynamic is flipped on its head as the performer becomes part of a larger texture. Further, their level of control over the performance is notably reduced now that they are adherent to time stamps and a fixed media track. By introducing fixed media so late in the work, Volness has transitioned *Particle Wave* from a degree 2 to a degree 1 on the interactivity spectrum.

GLOSSARY

Audio Interface: An input unit that allows analog equipment like microphones and pickups to communicate with computers. Modern interfaces normally employ some form of USB connection, but Firewire is fairly common in older models.

Automation: A function present in digital audio workstations that allows parameter changes to be automatically executed by the computer. Automation can be applied to almost any parameter available in a DAW.

Bit Depth: The number of bits employed in digital audio. The more bits that are present in a signal, the more realistic that signal is capable of sounding. Bit depth primarily alters the resolution of dynamics that a signal has: a higher bit depth results in more amplitudes.

Buffer: A designated space where an audio processing language stores a sample of audio to be manipulated.

Bus: A routing mechanism employed in mixing consoles and DAWS used to route individual audio signals together to manipulate them as one. Busses are most often used to create submixes, or to send audio to a channel for a time sensitive effect like delay or reverb.

Channel: A space on a DAW or mixing board that houses an audio signal. Channels are generally in either mono or stereo, and are where adjustments are made to volume, routing, audio effects, et cetera.

Clipping: When an audio signal exceeds the voltage capabilities of the system it is being played back in. Clipping sounds like distortion that only occurs at the loudest parts of a signal.

Convolution: The practice of multiplying two audio signals together to create an aggregate signal. This can be used to creative ends, but is most commonly used in reverb units. Here, a signal is multiplied by an impulse response which creates the illusion of reverberation.

Compressor: An audio effect that allows a user to reduce the loudest dynamic and increase the softest dynamic of an audio signal. This is used to increase the overall level of a signal and to give a sense of coherence to an audio mix.

Condenser Microphone: A device that converts sound waves into an electrical signal via a capacitor. Requires phantom power to function, which is generally supplied over an XLR connection from a preamp.

Contact Microphone: Microphones that interpret vibrations from physical objects rather than from the air like “normal” microphones. Contact microphones are places on the surface of whatever sound generating object one wishes to capture.

DAW: Stands for Digital Audio Workstation. A digital program designed for the recording, editing, and playback of audio.

Delay: An audio effect that is best compared to an echo. An audio signal is recorded and played back at a consistent interval, often decaying over time.

Direct Input (DI): Used to connect an unbalanced/line level audio signal to a balanced/microphone level input. Practically, this usually manifests as a small box that takes a ¼ inch input and outputs it to an XLR connector, usually to allow instruments like guitars, basses, and synthesizers to a mixing board.

Dry: A dry signal is one that has not received any kind of processing (EQ, reverb, et cetera), as opposed to a wet signal which has been processed. It is common for plugins to have parameters for the wet/dry mix to allow for control over how present the effects are.

Dynamic Microphone: A device that converts sound waves into an electrical signal via a conductive coil wrapped around a diaphragm inside of a magnetic field. Does not require phantom power, most common in live audio settings.

Eighth Inch Connector: A smaller variant of the TRS connector, often referred to as an auxiliary or "aux" connector. Most often used in headphones.

Equalizer: A filter for adjusting the loudness of different frequencies in a signal.

Fader: A slider used for altering the level of an audio signal. Originally found on physical mixing boards, faders have been adapted to the digital realm and are present in all modern DAWs.

Fixed Media: Refers to a piece of music containing elements that are played back as a recording rather than being performed live. There are works for fixed media with live instrumentalists and works for fixed media alone (often for 4, 8, or 6 channels of audio).

Foley Sound: The act of recording sounds in a studio to be used in visual media to increase audio quality. In many cases, Foley sounds will not be recreated literally, but will be approximated for effect.

Gain: Best thought of as how hard a microphone is listening. A higher gain will give a stronger instrumental signal, but might pick up more room noise. A weaker gain will eliminate room noise, but at the cost of the instrumental signal.

Graphical User Interface (GUI): The visual representation of a computer system that the user interacts with.

Granular Synthesis: A method of synthesis pioneered by composer Iannis Xenakis where incredibly short samples are played rapidly to create the illusion of a continuous signal. Granular synthesis can also be employed to creative ends to make drastic alterations to existing sounds.

Headroom: The space between the loudest point of a signal and the point at which that signal would clip. When setting headroom, a user should set the gain so that the signal is strong but there is enough space left over to allow for unexpected spikes in signal.

Instrument Level Signal: The signal level between microphone level and line level. Most common to direct output signals from instrument pickups, as in electric guitars or violin pickups. To convert an instrument level signal to line level, a preamp is employed.

Impulse: The audible result of a single percussive sound being made in a space. In audio, the attack of an impulse is often removed and the remaining sound is convoluted with a signal to model reverb.

In Line Processing:

Input: Any signal that is being fed into an audio system.

IRCAM: Stands for *Institut de Recherche et Coordination Acoustique/Musique*. IRCAM is a French research center where a vast majority of early experimental electronic breakthroughs were made, and is actively providing space and equipment for experimental music today.

Latency: Unlike analog audio, digital audio is not an immediate process; computers take time to process and convert signal

Line Level Signal: The signal level between mic/instrument levels and speaker level. To convert a microphone or instrument level signal to line level, a preamp is employed.

Live Electronics: When a musical work has an electronic (usually digital) component that is not recorded and played back. Here, a computer or piece of hardware is performing a function in real time during a performance, either listening and reacting to a performer or generating sound by some other, unrelated means.

Microphone Level Signal: The signal level that microphones operate at. This is the weakest level of signal, and requires the use of a preamp to become audible.

MIDI: Stands for musical instrument digital interface. MIDI compatible instruments and computers first appeared between 1981 and 1983. The protocol provides a common language for digital instruments and computers to communicate. Early MIDI devices made use of a five pin cable, but most modern devices send MIDI over USB.

Mixer: A control surface found in almost any recording studio or live music venue. Mixers are used to edit and balance live audio signals in order to create a coherent audible product. Can also be referred to as a console or board.

Mono: An audio signal that contains a single channel. This was common in early consumer recordings.

Mute: Temporarily turns off a signal, which can be useful when listening for balance between specific instruments in a mix. It can be found on channel strips in almost DAWs, hardware mixers, et cetera.

Object: The fundamental building block of object-based coding languages. Objects perform one specific task, and can be chained to other objects to create more complex systems.

Open Sound Control (OSC): An open source protocol for controlling audio parameters. Functionally similar to MIDI, but more adaptable.

Output: Any signal that is being fed out of an audio system.

Pan: In a stereo signal, balance of a signal between the left and right channel. In practice, this is used to give a signal a sense of location in space.

Parallel Processing: A term that refers to signal flow. To mix in parallel is to send both a raw, unprocessed signal to both the main output and to a submix, where it is further processed (compression, EQ, et cetera) and then sent to the main output. This is considered parallel because two versions of a signal are send to the output at the same time.

Patch/Patcher: In musical programming languages, a patch usually refers to the program designed by the user.

Phantom Power: A signal run over an XLR connector to provide power to a condenser microphone (without which the microphone will not function)/ Phantom power is found in preamps, and might be labeled as 48v in reference to the forty-eight volts of electricity being sent to the microphone.

Pickup: A transducer that converts vibrations into an audio signal. Most often found in string instruments like electric guitars/basses.

Polar Pattern: The direction(s) a microphone listens in.

Preamp: Short for preamplifier, sometimes casually referred to as a “pre”. A device that converts the relatively weak signal coming from a microphone to a line level signal that can be sent to a speaker system or power amplifier.

Quarter Inch Cable: See TRS/TS cable.

Reduced Listening: A concept pioneered by Pierre Schaeffer, where a listener rejects all preconceived cultural associations they may have with a sound. Instead, they listen exclusively to the timbral quality of the sound.

Reverb: An audio effect that gives a sense of space to an audio signal. Reverb is either achieved by running a signal through a metal plate or spring, or by convoluting it with the impulse from a specific space.

Ribbon Microphone: A microphone that operates by suspending a thin piece of metal (usually aluminium) between the poles of a magnet. Ribbon microphones are famously delicate, and most often have a bidirectional polar pattern.

Sample: A prerecorded (usually fairly short) audio recording that is used as a sound source for MIDI instruments or playback. In popular music, samples can also refer to sounds from other artists' work that are recontextualized in a new way.

Sample Rate: A term used in digital audio to describe the number of samples played back every second to simulate an analog audio signal. For example, a sample rate of 48,000 Hz contains 48,000 samples for every second of playback.

Send: A secondary output on an audio channel. Allows a user to send a signal to an auxiliary (aux) channel for further processing.

Speaker Level Signal: The signal level above line level. Speaker level signals are fed to loudspeakers and operate at a much higher voltage than line, instrument, or microphone level signals. As such they are carried on specific speaker cables (TRS) or Speakon cables.

Speakon: An audio connector used most often in live audio settings.

Splice: A term that originated in the days of magnetic tape. To splice audio is to cut it apart and connect it with another recording. This technique is used practically to edit recordings to eliminate mistakes, or creatively to create sounds that are not possible in the real world.

Solo: A function that singles out a chosen channel, which can be useful when working on specific signals/groups of signals. Multiple channels can be soloed at once. This function can be found in almost DAWs, hardware mixers, et cetera.

Stereo: An audio signal with two channels, left and right. The majority of modern consumer audio is stereo. The advantage of stereo audio is that instruments can be given their own space in a mix from left to right, lending a mix a more spacious and realistic feeling.

Submix: To mix a small group of instruments in a larger mix. This is accomplished by routing a group of instruments to a bus output and editing them as a whole before routing them to the main output.

Synthesis: The act of generating an audio signal by electronic means. Synthesis can be either analog or digital, and be managed by a number of different controllers like sequencers and MIDI controllers.

Tape Music: Music that is performed with magnetic tape that has been edited or manipulated in some way. Can describe either fixed media music (music with no performer) or music with performers playing along to the tape. In modern contexts, it is sometimes used to describe fixed media music on digital mediums.

Transport: Originally referring to devices used to control analog recording or playback devices, transport today usually refers to the digital panel where a user plays pauses, stops, rewinds, or fast forwards a recording or MIDI file.

TS: Stands for Tip Sleeve. They are sometimes referred to as instrument, unbalanced, ¼ inch cables. These cables feature a mono connector and are commonly used in pickups for string instruments (like electric guitar or bass).

TRS: Stands for Tip Ring Sleeve. They are sometimes referred to as speaker cables or balanced cables. They look identical to TS cables save for an extra ring around the input. Unlike TS cables, they are able to send either mono or stereo signals.

USB: Standing for “Universal Serial Bus”, USB is the most standard way to connect external hardware to personal computers. There are several variants of USB (USB A, B, C, Mini, Micro, etc). These connectors are most often encountered on external hard drives, audio interfaces, and in some cases microphones.

Wet: A wet signal is one that has received some kind of processing (EQ, reverb, et cetera), as opposed to a dry signal which has not been processed. It is common for plugins to have parameters for the wet/dry mix to allow for control over how present the effects are.

XLR: The standard connection used to send microphone signals. XLR stands for "External Line Return", and can be recognized by the three pins in a triangular formation. XLR cables have a "male" end for sending a signal and a "female" end for receiving one.

BIBLIOGRAPHY

Applebaum, Mark. *Aphasia*. Menlo Park, CA: Self Published, 1987.

Applebaum, Mark. *Pre-Composition*. CD. *Intellectual Property*. Innova, 2010.

Boulez, Pierre. *Anthemes 2*. Vienna, Austria: Universal Edition, 1997.

Cycling '74. “Max/MSP/Jitter.” Program documentation. *https://Docs.cycling74.Com/max8*. Cycling '74, September 25, 2018. https://docs.cycling74.com/max8.

Davidovsky, Mario. *Synchronism No. 6*. Philadelphia, PA: J.W. Pepper, 1970.

Dodge, Charles, and Thomas A. Jerse. *Computer Music: Synthesis, Composition, and Performance*. Second Editioned. New York, NY: Schirmer Books, 1997.

Durupt, Laurent. *Spirales*. Paris, France: Self Published, 2017.

Ellis, Brian. *My Battery Is Low And It Is Getting Dark*. Brooklyn, NY: Self Published, 2019.

Evans, Caleb. *Conversation*. Fayetteville, AR: Self Published, 2020.

Field, Ambrose. “Simulation and Reality: the New Sonic Objects.” Essay. In *Music, Electronic Media, and Culture*, edited by Simon Emmerson, 36–55. New York, NY: ROUTLEDGE, 2016.

Holmes, Thom. *Electronic and Experimental Music: Technology, Music, and Culture*. 6th ed. New York, NY: Taylor and Francis, 2016.

Huber, David Miles, and Robert E. Runstein. *Modern Recording Techniques*. 8th ed. Burlington, MA: Focal Press, 2014.

Irrational Games. *Bioshock*. 2k Games. PC/Mac. 2007.

Jones, Andy. “The Beginner's Guide To DAWs - The Basics.” MusicTech, June 21, 2017. https://www.musictech.net/guides/essential-guide/essential-guide-daws/.

Kosak, Dave. “The Beat Goes on: Dynamic Music in Spore.” GameSpy, February 20, 2008. http://pc.gamespy.com/pc/spore/853810p1.html.

Kuny, Ted. “A Digital Dark Ages? Challenges in the Preservation of Electronic Information.” *Audiovisual and Multimedia Joint with Preservation and Conservation, Information Technology, Library Buildings and Equipment, and the*

PAC Core Program. Lecture presented at the 63RD IFLA Council and General Conference, September 4, 1997.

Landy, Leigh. *Understanding the Art of Sound Organization*. Cambridge, MA: MIT Press, 2007.

Lansky, Paul. *Idle Chatter*. CD. *More Than Idle Chatter*. Naxos Digital Services US Inc, 1994.

Lillios, Elainie. *Time-Lapse*. Bowling Green, OH: Self Published, 2021.

Manzo, V. J. *Max/MSP/Jitter for Music*. New York, NY: Oxford University Press, 2016.

Martínez, José. *Monologue V*. Austin, TX: Self Published, 2021.

Matalon, Martin. *Traces II*. Paris, France: Éditions Billaudot, 2005.

Miller, Tyler. "The Speaker Is Present: A Conversation with Lesley Flanigan." IMPOSE Magazine, March 29, 2016. https://imposemagazine.com/features/lesley-flanigan-hedera-interview.

Moles Abraham André. *Les Musiques expérimentales: Revue D'une Tendance Importante De La Musique Contemporaine*. Paris: Editions du Cercle d'Art Contemporain, 1960.

Moore, Adrian. *Sonic Art: an Introduction to Electroacoustic Music Composition*. New York, NY: Routledge, 2016.

Olson, Dan. *Ludonarrative Dissonance*. *Youtube*, 2017. https://www.youtube.com/watch?v=04zaTjuV60A&t=139s.

Parker, James. *Baptism of Wind and Waves*. Austin, TX: Self Published, 2020.

Payne, Matthew Joseph. *Flight of the Bleeper Bird,* Self Published, 2013.

Perich, Tristan. *1-Bit Symphony,* Physical Editions, 2010.

PQ, Rory. "What's the Best Effects Chain Order for Mixing?: Icon Collective." Icon Collective College of Music, May 4, 2020. https://iconcollective.edu/mixing-effects-chain-order/.

Rayburn, Ray A. XLR History. Sound First, December 20, 2017. http://www.soundfirst.com/xlr.html.

Reich, Steve. *Different Trains*. New York, New York: Boosey & Hawkes, 1988.

Reich, Steve. *Electric Counterpoint*. New York, New York: Boosey & Hawkes, 1987.

Roads, Curtis, John Strawn, Curtis Abbot, John Gordon, and Philip Greenspun. *The Computer Music Tutorial*. Cambridge, MA: MIT Press, 2012.

Schaeffer, Pierre, and Christine North. “Translator’s Notes.” In *In Search of a Concrete Music*, translated by Christine North and John Dack, xii. Berkely, CA: University of California Press, 2013.

Schaeffer, Pierre. *Treatise on Musical Objects: Essays across Disciplines*. Translated by Christine North and John Dack. Oakland, CA: University of California Press, 2017.

Smith, Jonathan Andrew. *Inquietude,* Self Published, 2020.

Stockhausen, Karlheinz. 1962. "The Concept of Unity in Electronic Music (Die Einheit der musikalischen Zeit)". Translated by Elaine Barkin. *Perspectives of New Music* 1, no. 1 (Autumn): 39–48.

Stockhausen, Karlheinz. *Kontakte*. Kürten, Germany: Stockhausen Verlag, 1960.

Varèse, Edgard. “The Liberation of Sound.” Essay. In *Audio Culture: Readings in Modern Music*, edited by Christoph Cox and Daniel Warner, 17–21. New York, NY: Bloomsbury Academic, 2017.

Varèse, Edgard. *Poème électronique.* New York, New York: Boosey & Hawkes, 1987.
Volness, Kirsten. *Particle Wave.* Self Published, 2020.

Westlake, Nigel. *Hinchenbrook Riffs*. Sydney, Australia: Rimshot Music, 2003.

Wharton, Russell. *Deus Ex Metronome*. Portland, Oregon: Tapspace Publications, 2019.

Windsor, Luke. “Through and Around the Acousmatic: the Interpretation of Electroacoustic Sounds.” Essay. In *MUSIC, ELECTRONIC MEDIA AND CULTURE*, edited by Simon Emmerson, 7–35. New York, NY: ROUTLEDGE, 2016.

Yager Development. *Spec Ops: The Line*. Epic Games. PC/Mac. 2012.