Tasteful DSP: How Can Compositional Desires be Embedded into Live Electronics Systems?

Jack Walker

Edinburgh College of Art
The University of Edinburgh
jackwalker26@live.co.uk

Abstract. This paper explores a specific mapping problem. I have a corpus of sounds and a musical intent for a corpus-based piece of live music, but do not have an interface that is able to transmute these low-level sound materials into higher-level musical form. While I can identify readymade solutions to this problem, I have no assurance that I will like the sound that will emerge from these tools. What steps can I take to design a musical interface that is likely to produce things I enjoy? I solve this problem by interrogating my own assumptions about how this tool should behave, before designing a process that can realize this musical potential. I conclude by arguing that interesting musical affordances occur when we consider behavioral commonalities of humans and machines and that this correspondence can help us to embed our musical desires into live electronics systems.

Keywords: machine listening, AI, live electronics, improvisation, cybernetics

1 Introduction

"The practice of building new musical instruments is predicated on the recognition that instruments matter: that the sort of music one can make with a xylophone is different than with a violin, which is different still from the music one can make with a computer. Instruments differ by more than just their sound qualities; acoustic instruments bring with them particular physical affordances, and these lead to idiomatic playing styles and repertoires." (Fiebrink, 2017, p. 01)

Rebecca Fiebrink's claims (above) neatly encapsulate the conceptual thrust of this paper. New modes of interacting with sound give rise to new idioms, techniques and repertoires. As a composer of pieces for computers and people, I want to better understand how I can integrate the affordances provided by machine listening into my composition work. My aim in doing so it to establish a design strategy for DSP software tools that can evaluate input audio data and autonomously produce parameter mappings that musically respond to input variance. While this is computationally a simple task (with solutions for doing so provided by Fiebrink's Wekinator (2017), Benjamin D. Smith's ml.star (2017), FluCoMa (n.d.) and Ircam's MUBU (2021)), there is a key aesthetic problem: if my system is given an unpredictable sound input, how can I ensure that I will like the musical result?

I use a specific mapping problem as a way of exploring this idea. In 2019, I took part in an artist residency with the Institute for Electronic Music's Inter_Agency research project (IEM, n.d.; see *Section 2* for more details) During this time, I accumulated a large corpus (I use this term to refer to a collection of sound files. See, Tremblay, Green and Roma, 2019 for a more nuanced discussion of musical corporea) of audio data. I want to use my guitar as a way of provoking musical responses from this corpus. To do this, I will need to design an interface. I begin this paper by giving an overview of my mapping problem. I then discuss certain theoretical and practical assumptions that underpin my attempt to strategize an appropriate interface design. Finally, I provide a solution to this problem, by theorizing on how my dilemma can be reconciled into a neat and legible interface for musical exploration, that can then be further tidied up into a more sophisticated system that is capable of mediating the electronic form of entire works of music. My argument is supported by examples of my own composition work, please follow this link to find the source folder. You will be directed to musical examples when they help to evidence key claims.

2 My problem: a cold and stationary corpus of music that I like

As an artist with residence with the Institute for Electronic Music's Inter_Agency research project, I was invited to experiment with "intelligent agent-based systems and the application of machine listening and machine learning in interactive electro-instrumental compositions" (IEM, n.d.) Working with double bass player Maggie Maierhofer-Lischka and percussionist Manuel Alcaraz of Graz's Schallfeld Ensemble (n.d.), I began composing a work of music for live electronics and the two players. Core to my creative intent was the idea that musical ideas can exist as abstract forms, capable of being shared between and acted upon between human beings and machines. The human and my computer system were encouraged to listen to the musical behavior of their co-players (a term that I use to include my computer) and respond in a semi-structured manner (details below), so that each all players – human and nonhuman - contribute to the emergence of a group improvisation.

2.1 My musical intent

I approached this by conceptualizing my piece's 'score' as a set of abstract instructions that govern how the humans and computer could musically behave. Equivalence in messaging was key here: I wanted to feel like I was using the same vocabulary to direct both parties. Further detail on this will be provided below (see *Section 2.2* and *Section 4*), but a simple example of this can be found in the word 'listen'. While the process of 'listening' to sound will be different depending on whether you are a biological organism or a robot, we can use the term 'machine listening' (Gioti, 2017; Bowers and Green, 2018) to understand what happens when a computer extracts sound information from a real-time audio input. If I understand 'listening' as an umbrella referent for both biological *and* machine listening, then I can find a single abstract instruction that can meaningfully address both humans and machines.

My piece was never finished and performed in concert due to the Covid-19 pandemic. While this was quite disappointing (although starkly trivial in face of the pandemic's global impact), on closer inspection it gave me an opportunity to interrogate my musical intention with an air of sobriety not easily afforded by the chaotic (mis)management of coding, rehearsals, caffeine and anxiety that usually accompanies the final stages of a work of my concert music. With the small amount of time that I had to finish the piece (roughly two weeks) suddenly expanded to outstretch the length of this global health crisis, I had the opportunity to further unpack my desires about the project. As will become abundantly clear throughout this paper, this task of understanding what I wanted from a piece of music is not a trivial endeavor.

2.2 My corpus

I recorded four multitracked rehearsals with Manu and Maggie, resulting in around 20 hours of raw audio data. In each rehearsal, we discussed the types of sound that we were interested in and then experimented with some short semi-directed improvisations. In these mini explorations, I would invite the player to use a particular playing technique (i.e. a specific preparation for Maggies's bass or a particular set of percussion for Manu's set), or to move between between a few different sound behaviors. After each short improv, we discussed the music and how we felt about it. We quickly began to a small set of modes of making predictable improvised sound (i.e. 'fast modes'; 'tonal modes'; 'modes with multiphonics'; 'modes with groove'; modes where the players mimic/counterpoint each other'). From these specific methods of improvising sound, we devised a shortlist of modes that we felt worked well together. If I directed the players to one of these modes, I was largely able to predict the sounds that I would hear. This gave us a shared and abstract language for communicating about how we could structure the piece. (Please see here for recordings of these sessions.)

My next challenge was to refine a set of synthesis parameter presets that I felt were sonically alike to these modes of playing, so that every modes could be reliably achieved by the humans and the computer. I devised a system for switching between preset states in accordance with sound description data taken in and classified by Wekinator (Fiebrink, 2017) – a tool that can be used to trained to classify multiple different types of sound behavior. The tension in the piece would then arise when Wekinator classified input data 'incorrectly' (a term that I use with a caveat – any misclassifications were likely to be my error in training Wekinator, rather than a mistake made by the machine). This would encourage moments where the electronics and the ensemble would contradict each other, resulting in a more chaotic order of events. The players could then sustain this tension or fallback into easy beds of sound that the machine would have less difficulty classifying to retrieve a sense of unity and cohesion (see Bowers and Green, 2018; Green, 2013 for discussion of how 'errors' in machine listening algorithms can produce musically interesting results). These fallback states would be communicated to Manu and Maggie ahead of time, in manner similar to the 'security measures' deployed by performers to avert 'emergency situations' (e.g. specific unwanted system behaviors) in Agostino Di Scipio's Background Noise Study (2005, p. 14).

2.3 My problem

My problem is that this never happened. I am trapped in an open-ended phase of composition that is never able to collapse into a singular reality through performance. I have fantasies about what the music could be rather than clear directives for achieving this potential. However, I have recordings of the rehearsals. This corpus of sound clearly reflects the types of sound behavior that I found so exciting in the formative explorations of the piece and I believe contains sufficient musical information for a full work of music. The question then becomes how I can use these lower-level sound materials as a way of producing a piece of music that is representative of what I wanted to get out of the project. I therefore have a very specific mapping problem to solve: I have a corpus of sound that I need to extract musical fragments from, so that these fragments can be reconstituted into a structure that I enjoy listening to. To do this, I need a clear strategy for (a) extracting useful parts of material, (b) reassembling these forms into new musical events in a way that (c) is reflective of my musical intent and desires for the piece. As an improviser myself - and wanting to continue the live musicking (Small, 1998) element of the project - I decided to design a system that takes an input from my guitar, analyses it and then attempts to playback a selection from my corpus that is roughly correlated to the input data (see *Section 4* for more information on how this can happen).

3 How do I want to solve this problem?

As stated, I have a corpus of musically viable material that I want to be able to interact with by playing a guitar. Before moving onto how I could solve this problem by designing a musical interface, I set out some principles that undercut my design strategy: three assumptions about music practice that are inherent to my understanding of how I can make music. First, music is a relational phenomenon, irreducible to a singular conception of material| (*Section 3.1*). This used to point towards the conceptual intent of my design strategy and ask what it is that I am making. Second, music emerges from ecologically situated interactions occurring between various discrete entities (*3.2*). This assumes that I can approach music composition by designing a 'system', which in itself positions my work into a wider practical context. Third, the lower-level components (the aforementioned discrete entities) that constitute my music system can be approached as if they are able to make decisions that are independent to mine (*3.3*).

3.1 Music is materially problematic

"(M)usic has no material essence but a plural and distributed material being.....music's multiple simultaneous forms of existence—as sound, score, discourse, site, performance, social relations, technological media—indicate the necessity of conceiving of the musical object as a constellation…Compared with the visual and literary arts, then, music has to be grasped as an extraordinarily complex kind of cultural object—as an aggregation of sonic, visual, discursive, social, corporeal, technological, and temporal mediations (Born, 2017 p. 44).

Musicking is a complex set of different relational factors, irreducible to a single human behavior or a specific conception of material. I draw attention to Georgina Born's claim (above) for two reasons. First, it heightens the socially relational elements of the piece. Although I am directing this project towards a solo work, I cannot sufficiently underline just how collaborative this process remains. I was explicit with the players about how much I valued their contributions; the piece emerged from my social interactions with these highly creative and talented individuals. My interface design will need to sonically factor in this socially relational component (Davis, 2010; Born, 2017). This is a matter of principal – I would feel uncomfortable about transforming the sound from my corpus in such a manner that the ensemble's contributions became masked by my processing (e.g. if I was to granulate and reconstitute the sounds to such a microsonic extent that a third-party listener could not realize that I started off with expertly played acoustic musical instruments). Second, this relationality of musical material raises a design problem. In such a diffuse environment, where one must account for such a complex range of interrelating mediatory features, where does the composition work take place?

3.2 Music is ecosystemic

We can answer this question with reference to Simon Waters' *performance ecosystem* (2007). Waters problematizes conventional art music practices that reify hard boundaries between the concepts of *player*, *instrument* and *environment*. While any mutability between these categories can be unproblematically elided when considering the Western classical tradition – where one could easily decode the distinctions between a *guitarist*, their *guitar* and a *concert hall* – Waters fixates on performance practices that cannot be explained in a way that reconciles this assumed separation of elements. Sound becomes an *emergent* phenomenon. In contrast to practices where music is encoded to a relatively high level of detail prior to performance (e.g. fixed media acousmatic music or karaoke), ecological sound oozes contingently from complex and unstable relationships between people, technologies and spaces (Waters, 2007; Green, 2008; Davis, 2010; Green, 2013; Furniss and Parker, 2014). This ecosystemic nature of music practice allows us to approach composition by consciously designing a complex socio-technological system. This affords a language for thinking through music as a model of environmentally situated interactional behavior (Impett, 2001; Di Scipio, 2003; Waters, 2007; Pickles, 2016; Sanfillipo, 2017).

This discursive framework is complimented by a constellation of musical approaches that set this idea into conscious motion. Notably, this includes Agostino Di Scipio's audible ecosystems: works of music that adapt themselves sonically to their surroundings (2003; 2005). Moreover, these ideas are critically and practically extended (deliberately or otherwise) by the efforts of Owen Green (2008; 2013), Artemi-Maria Gioti (2019), (Tina Krekels, 2019) Alice Eldridge (Kiefer and Eldridge, n.d.), Chris Kiefer (*ibid*), Dario Sanfillipo (2017), Martin Parker (2013), Pete Furniss (Furniss and Parker, 2014), Pierre-Alexander Tremblay (Tremblay and Schwarz, 2010) and Diemo Schwarz (*ibid*), among other composers who fixate upon these interactions between people, technologies and places. My musical interface design is not a solipsistic endeavor – it fits into a wider landscape of musical practices and discursive approaches that share aims, problems, techniques and solutions. As a practice-based researcher, I want my composition work to add something useful to this community (see Bown, Eldridge and McCormack, 2009, for discussion of how practitioners connect with their wider musicking community and Green, Tremblay and Roma, 2018, for a specific approach to generating this collective (cyber)space for sharing, debating and developing electronic musical practice).

3.3 Music systems can contain multiple agencies

The technological entities that constitute these music systems (pieces of software or hardware that perform a specific interactive function within the context of the system) can act as though they have some musical agency that is separate to that of the system's composer (Di Scipio, 2003; Bown, Eldridge and McCormack, 2009; Gifford et al, 2018; Gioti, 2019) The notion that computationally invoked agency – a concept familiar to artificial intelligence - can have relevance to music practice is an uncontroversial claim, given the nature of the AIMC proceedings that contain this paper (for a deeper exploration of AI in music research see Gioti, 2020, and the remainder of this conference proceedings). However, the extent of this agency that is held by technological components is subject to enormous variance, subject to the composer's specific technical design. Gifford et al (2018) provide one robust approach to critiquing the extent of agency that is exhibited by a computer music system with their concept of 'creative agency'. Their understanding of the agency that is demonstrated by music systems rests upon whether the human being can understand the system "as contributing to the ongoing creative collaborative activity with some degree of autonomy" (*ibid,* p. 1) potentially from "emergent, complex dynamics within the system's design, or…algorithms designed to instill autonomous behavior into the system." (pp. 2-3)

Their study reveals a spectrum of approaches to achieving this, spanning from those featuring "heavy use of hard-coded rules of musical structure and/or preprogrammed sequences" (p. 13), and freer systems that "impose little stylistic constraints on the performer, but rely on human listening as the primary aesthetic evaluation method" (*ibid*). From this, we can begin to trace different ways in which composers can embed their own compositional preferences, stylistic tropes and musical desires onto machines. Systems from the 'hard-coded' end of the spectrum will potentially assume that their human counterparts will be able to understand and respond sensibly to the particular stylistic idioms and tropes that are embedded into the software (presumably mapped according to the composer's personal tastes), while the latter may adapt more fluidly to the environment into which it is deployed, allowing the human player to project their own musical preferences onto the machine with less real-time resistance. The diversity of this taxonomy (and the range of musical approaches taken by the practitioners listed in the *Section 3.2*) deepens my problem. With such plurality on offer, how should I determine where my musical will fall on this spectrum? In the following section, I sketch out a strategy for answering this question.

4 My solution: abstraction

My core problem is that I have a corpus full of interesting sounds, a musical intention for a piece that explores using abstract instructions to guide both humans and machines, but no interface for realising this musical potential. In the following section, I sketch out an approach that I could take as a way of solving these problems. In doing this, I prepare a strategy for learning what I want my interface to do and set out some clear pathways for allowing this to happen. As a designer of systems that feature a strong technological component, part of my practice as a composer is the design of technology (see Green, 2008, for an insightful discussion of technology as practice in live electronic musicking). When thinking through the technology that I will need to design in future, I can towards technological approaches that have worked out for myself and other practitioners in the past. Given that the crux of this research is an exploration of my own desires, it seems fitting to begin with an assessment of whether any of my past musical interfaces can be broken down into reusable components.

4.1 Simple implementations of machine listening

I first discuss a short project that I carried out towards the start of my residency with IEM's Inter_Agency team (IEM, n.d.; further details provided in *Section 02*). To get acquaint myself with the project's aims, I tried to implement the simplest possible machine listening music systems. I designed a system that varies the frequency of a sine wave in accordance with continuous changes of input amplitude (an *envelope follower* system). Another sets the frequency of a sine to a random value whenever the amplitude exceeds a certain threshold (an *event detector* system). While these systems showed some awareness of their environment (reduced to the single dimension of input amplitude) and were able to respond by modulating a musical parameter (pitch), there was a distinct lack of satisfying and complex interaction.

Predicting the behavior of the envelope follower was trivial: loud inputs increase pitch; quietness does the opposite. The event detector was less determinate, due to the randomness of output pitch values. However, the interaction was similarity unrewarding: I make a loud sound and I receive a random pitch in response. In order to implement more interesting behaviors, I needed to scale up the complexity of my designs. I implemented an interface with two envelope followers, two microphones and a loudspeaker. Sound was fed from microphone array to loudspeaker to produce audible Larsen tones. The envelope followers dampened the sensitivity of the system: amplitude received into the left-hand microphone reduced the level of the right hand microphone input and vice-versa. This allowed tangible control over the audible feedback. I could scratch the casing of either microphone to reduce the chance of the other mike feeding back. This system 'felt' like it had some crude musical agency (see *Section 3.3*). It was difficult to predict the way that sound would emerge, but there was a clear deterministic logic to sound production that I was able to learn through a small amount of practice. This complexity was achieved through a feedback loop: when the system was able to respond to its own outputs – the manner in which it modified the acoustic of its host environment - the interactional complexity scaled up significantly.

With a tiny amount of coding, a laptop connected to an audio interface, two microphones and a loudspeaker, I was able to use my body as a way of exploring the sonic possibilities afforded by a complex feedback instrument. This tiny infrastructure was easily integrated with existing music systems. First, I played my feedback instrument while interacting with Tom Mudd's Gutter Synthesis software (Mudd, n.d.). For this improvisation, I set myself the challenge of using my ears to track the pitch of the emergent larsen tone, so that I could attempt to match the parameters of the Gutter interface to encourage a harmonic relationship between the two sound sources. The sounds from the Gutter were fed back into the microphone and reinforced the Gutter-like qualities to the Larsen Tone generator. Second, I loaded up an audio-driven corpus-based concatenative synthesizer (see Tremblay and Schwarz, 2010), pilfered and quickly adapted from the MUBU MaxMSP documentation (IRCAM, 2021). This system extracted sets of real time sound descriptions from a sum of my microphone inputs and searched through its corpus of sounds to find a short snippet of sound with the most similar descriptor profile to the most recently evaluated input dataset. Again, this produced complex self-reinforcing behaviors, as sounds from my corpus (which was filled with samples of myself improvising on guitar) could be heard by my microphones, meaning the concatenation system would fall into brief patterns of responding to its own output. Neither of these systems were labor intensive from a coding perspective: it was possible for me to vastly increase the technological complexity of my system with very little additional code. Please see here for some recordings of my three machine listening interfaces.

4.2 Extractable components and extensible design

As I look towards designing my corpus-based musical interface, there are a few core ideas that can be pulled from these extant interfaces for improvising sound. First, my first experiments reveal a design ontology. When analyzed with a suitable level of abstraction, the concept of 'event detector' works just as well for describing my crude attack detector as my corpus-based concatenative synthesizer, despite the fact that the technical complexity of these tools is very different. Each detector receives an audio input, extracts specific information the sound the sound (amplitude onsets in the random sine wave generator and a multidimensional set of spectral descriptors in the concatenative synthesizer), and then use this information to trigger a specific musical response (a random change in pitch or the algorithmic selection of a fragment of sound). Second, these abstractions can easily adhere to my specification for a control language that can be communicated to both people and machines. While I initially started using the term 'envelope following' to refer to the actions of a computer, my experiment with the feedback instrument and Gutter synth reveals an instruction for biological envelope following: '*track the changing envelope of a pitched signal and respond my attempting to mimic it with sound*'. It would be simple enough to implement a biological event detector: '*produce a sound whenever they can hear me clap my hands*'.

Furthermore, there is a process that can be extrapolated from these low-level ways of exploring with sound. The focus here is on swift embodied design: the tools were produced very quickly and used as interfaces for exploring a particular set of musical affordances. This resonates with the affordances provided by Wekinator, which is a machine learning solution to producing arbitrary mappings between input and output very quickly and without requiring extensive coding abilities from its user (Fiebrink, Trueman and Cook, 2009). However, Fiebrink argues that machine learning can reduce the need for extensive coding even further: "(a)nother critical difference between designing instruments using machine learning and designing instruments by writing code is that composers are able to use their bodies directly in the design process." (Fiebrink, 2017, p. 6).

Similarly, Huddersfield's FluCoMa (n.d.; Green, Tremblay, and Roma 2018; Roma, Green and Tremblay, 2019; Tremblay, Green,. and Roma, 2019) project aims to put machine learning techniques into the hands of composers. The FluCoMa library offers machine learning solutions in environments commonly used by experimental music practitioners (Pure Data, MaxMSP and Supercollider). Simply by following the example patches provided by Fiebrink and FluCoMa, it would be very easily to quickly design interfaces that contain complex machine learning technology but retain simple and light-touch approaches to coding. and instead spend my time practicing with methods of retrieving musical events from my corpus and evaluating the how I feel about the sounds that I hear. If I do not like a specific approach, then I can easily discard it and pursue something else. This offers foregrounds the tasks of playing and listening to guide an intuitive process of compositional decision making (Cook, 2019); an embodied process of 'feeling out' my musical desires and locking them into interfaces for future exploration.

4.3 Risks of wrapping things up

There is a risk inherent to this approach: the simple and hacky interfaces that emerge from this process *may* be sufficient enough infrastructure for an entire piece of music, but this is not a guarantee. A problem that I had with my three quickly coded machine listening interfaces is that I quickly became quite bored with the opportunities and wanted the electronics to change. How can these lower-level tools for exploration become integrated into more complex systems for producing a greater variety of musical possibility? One solution could implement timed events to simply move the system through different parameter states. I have implemented this in the past when I needed tight synchrony between a scored work and live-electronics. However, this problem could be solved with machine listening tools: Martin Parker's GruntCount (2013) monitors input microphones for moments of loudness, and maps each of these attacks (or 'grunts') to move through a DSP parameter space, which is applied as processing to an improvisers instrument. If the musician plays very quietly, then the electronic component of the piece moves very slowly; if loud then fast (Furniss and Parker, 2014).

Again, this spectrum reveals further problems: with all this potential, how should I make this decision for my own piece? To stake out an approach to answering this, I analyze three questions that Owen Green (personal communication, June 2021) asked me when I discussed this piece with him. First, how long does the piece need to be? The code infrastructure needed to retain my interest for five minutes seems likely to be different to a system needed to run for forty minutes. Second, how repeatable does the piece need to be? If the piece needs to play in the same way every time, then I may want to orient my system around a fixed set of timed changes, rather than an interactive way of mapping through parameter spaces. Third, who takes responsibility for the way in which sound is formed throughout performance? As argued above, software components of music systems can project their own agency (see *Section 3.3*). Do I want this, or do I want to have more control over the procedural unfolding of sound events?

Given the sheer range of material approaches that I could take to structuring this work, Green's suggestion that I look inward as a means of finding these solutions is key to understanding how I can approach the design of my interface. Again, this task of looking inwards resonates with my use of abstraction. By analyzing the way in which I have produced music in the past, I can identify worthwhile features and processes contained within my design that are repeatable outside of the context that they originated in. This higher-level understanding of the components of my music systems allows me some degree of modularity when designing works of music. Crucially, this understanding of system components as high-level abstractions can lean towards a language of control that can be articulated to both human beings and machines. If we abstract music making away from the technology that affords it, and simultaneously away from the biological, social and cultural forces that propagates it, then we can address the act of creating music as a relational process. We can cohere a musical language that reconciles the mutability of player, instrument and environment. We can invite an understanding of what we want to do with this language by interrogating the specificity that we bring to this complex relational task of composing music by making systems.

4 Conclusion

I started this paper with an overview of a specific and ostensibly trivial mapping problem. I had a corpus of sound that I enjoyed listening to, a musical intent for a piece of music, but no clear directives for building an interface that can make this music. It should now be clear that the aim of this paper was not to solve this specific problem (a task that would be more fruitfully accomplished by playing music rather than pontificating about it). I have, however, provided a conceptual basis for weaving my compositional desires into the systems that I use to make music. My desires are not material in any way that I can make practical sense of - I cannot grasp them in my hand and throw them at musicians or computers in the hope that they coalesce into sounds that I enjoy hearing. Moreover, the act of discovering my musical desires is not as simple as thinking that I know what I want. For myself at least, it takes embodied musical practice, interpersonal relationships with other musicians, participation in a musicking community, experimentation with music technology, and self-awareness. This iterative process foregrounds self-query, play and exploration as a means of understanding myself as a composer. The specificity that I bring to music practice allows me to untangle this web of relational affordance into a specific interface for musical experimentation. Once they exist, these interfaces can be evaluated as containers of my musical desire; sources for an abstract control language that I can communicate to machines and other people.

References

Born, G. (2017). After Relational Aesthetics: Improvised Music, the Social and (Re)Theorising the Aesthetic. In Born, G., & Reinharz, J. (Eds) *Improvisation and Social Aesthetics*. Durham: Duke University Press.

Bowers, J. and Green, O. (2018). All the Noises: Hijacking Listening Machines for Performative Research. *Proceedings of the Conference on New Interfaces for Musical Expression 2018*.

Bown, O., Eldridge, A. and McCormack, J. (2009). Understanding Interaction in Contemporary Digital Music: From Instruments to Behavioural Objects. *Organised Sound* 14(2): 188–196, Cambridge University Press.

Cook, N. (2018). *Music as Creative Practice*. Oxford University Press.

Davis, T. (2011). Towards a Relational Understanding of the Performance Ecosystem. *Organised Sound*, 16(2), 120-124, Cambridge University Press.

Di Scipio, A. (2003). Sound is the Interface. *Organised Sound* 8(3): 269–277, Cambridge University Press.

Di Scipio, A. (2005). Audible Ecosystemics n.3a (Background Noise Study). Retrieved from: tallersonoro.com/anterioresES/13/Audible_ecosystems_full_score.pdf

Fiebrink, R. (2017). Machine Learning as Meta-Instrument: Human-Machine Partnerships Shaping Expressive Instrumental Creation. In T. Bovermann et al., Editors, *Musical Instruments in the 21st Century*. pp. 137–151. Springer.

Fiebrink, R., D. Trueman, and P. R. Cook (2009). A meta-instrument for interactive, on-the-fly machine learning. *Proceedings of New Interfaces for Musical Expression (NIME)*.

FluComa (n.d.). *Fluid Corpus Manipulation*. Retrieved from: flucoma.org/about/

Furniss, P. and Parker, M. (2014). gruntCount: Blurring the Piece/System/Instrument distinction. *Proceedings of the Inter-Face: International Conference on Live Interfaces*

Gifford, T., Knotts, S., McCormack, J., Kalonaris, S., Yee-King, M., & d'Inverno, M. (2018). Computational systems for music improvisation. *Digital Creativity*, 29(1), 19–36.

Gioti, A. M. (2017). Machine Listening in Interactive Music Systems: Current State and Future Directions. *Proceedings of the 2017 International Computer Music Conference*.

Gioti, A. M. (2019). Imitation Game: An Interactive Composition for Human and Robotic Percussionist. *Proceedings of the 2019 International Computer Music Conference.*

Gioti, A. M. (2020). From Artificial to Extended Intelligence in Music Composition. *Organised Sound* 25 (1): 25-32, Cambridge University Press.

Green, O. (2008). Pondering Value in the Performance Ecosystem. Retrieved at: econtact.ca/10_4/green_ecosystem.html

Green, O. (2013). User Serviceable Parts: Practice, Technology, Sociality and Method in Live Electronic Musicking. Unpublished PhD Thesis, London: City University London).

Green, O., Tremblay, P. A. & Roma, G. (2018). "Interdisciplinary Research as Musical Experimentation: A Case Study in Musicianly Approaches to Sound Corpora". *Proceedings of the Electroacoustic Music Studies Network Conference*.

IEM (n.d.) *Inter_agency*. Retrieved from: interagency.iem.at

Impett, J. (2001). 'Interaction, Simulation and Invention: A Model for Interactive Music' in Bilotta, Miranda, Pantano and Todd (eds.) *Artificial Life Models for Music Applications*, pp.108-119. (Cosenza: Bios).

IRCAM (2021). MUBU. Retrieved from: ismm.ircam.fr/mubu/

Kiefer, C and Eldridge, A. (n.d.). Feedback Cell. Retrieved at: feedbackcell.info

Krekels, T. (2019). *Loosening The Saxophone: Entanglements of Bodies in the Politics of Free Improvisation*. Unpublished PhD Thesis. Edinburgh: University of Edinburgh.

Mudd, T. (n.d.). *Gutter Synthesis*. Retrieved from: entracte.co.uk/projects/tom-mudd-e226/

Parker, M. (2013). *gruntCount*. Retrieved from: sumtone.bandcamp.com/album/gruntcount-improvised-pieces-for-player-and-computer

Pickles, D. (2016). *Cybernetics in Music*. Unpublished PhD Thesis. Coventry: Coventry University.

Roma, G., Green, O., & Tremblay, P. A. (2019). *Fluid Corpus Map: Adaptive Mapping of Sound Collections for Data-driven Musical Interfaces*. Retrieved from: github.com/flucoma/FluidCorpusMap

Sanfilippo, D. (2017) Time-variant Infrastructures and Dynamical Adaptivity for Higher Degrees of Complexity in Autonomous Music Feedback Systems: The *Order*

From Noise (2017) Project. *Musica/Tecnologia*, (11) 121-131, Firenze University Press.

Schallfeld Ensemble (n.d.). Schallfeld Ensemble. Retrieved from: schallfeldensemble.com

Small, C. (1998). *Musicking*. Wesleyan University Press: Connecticut.

Smith, B.D. (2017). *ml.**. Retrieved from: benjamindaysmith.com/ml-machine-learning-toolkit-in-max

Tremblay, P. A., Green, O. and Roma, G. (2019). From Collections to Corpora: Exploring Sounds through Fluid Decomposition *Proceedings of the ICMC-2019*.

Tremblay, P.A. and Schwarz, D. (2010). Surfing the Waves: Live Audio Mosaicing of an Electric Bass Performance as a Corpus Browsing Interface*. Proceedings of the 2010 International Conference on New Interfaces for Musical Expression* (NIME10), Sydney, Australia. NIME, pp. 447-450.

Waters, S. (2007). Performance Ecosystems. *Proceedings of EMS*.