

Live Coding the Global Hyperorgan: The Paragraph environment in the indeterminate place

MATTIAS PETERSSON^{1,2} 

¹Luleå University of Technology, Piteå, Sweden

²Royal College of Music, Stockholm, Sweden

Email: mattias.petersson@ltu.se

This article presents several scenarios in which a live coding environment called Paragraph was utilised to telematically play networked and geographically distributed hyperorgans. Situated within the framework of the *Global Hyperorgan* project, the TCP/Indeterminate Place Quartet have explored the affordances of the organ network through the concept of Tele-Copresence. By outsourcing certain dimensions of the parameter space of the Paragraph language to other members of the quartet, a shared instrumentality is enabled, where the organs are collaboratively controlled by means of this system. Rooted in a personal composer-performer practice and studied from the perspective of the live coder, the Paragraph system, adapted to the TCP/Indeterminate Place environment, can be understood as a modular system of human and non-human agents, into which the other musicians are patched. The distributed parameter space utilised, thus resembles a shared *cantus firmus*, a foundational, but dynamically changing, ecology for the live coder to play within.

1. INTRODUCTION

As a singular live coder in a mixed ensemble, latency is a tangible issue and ‘[t]he time it takes between designing musical ideas syntactically, executing the code block and finally hearing the result is a defining part of the [live coder’s] instrument’ (Harlow, Petersson, Ek, Visi and Östersjö 2021: 6). This latency could be considered one of the strengths of live coding, while in some types of musicking the performer might feel the need to quickly adapt and react to fellow musicians (ibid.). Such needs became apparent in a project where several networked hyperorgans were live coded by the author within the TCP/Indeterminate Place Quartet. The number of such organs that can be remotely controlled by means of OSC and MIDI is constantly increasing, and these hyperinstruments therefore also afford novel possibilities for telematic performance. Using a live coding framework as a proxy, thus replacing the organ’s regular keyboard interface, opens up for new approaches to explore the inner modularity of the organ.

In a number of different performances made within the *Global Hyperorgan* project (Harlow et al. 2021),

the TCP/Indeterminate Place Quartet have explored the concept of ‘Tele-Copresence, as an expression of [their] central interest in sense of presence in musical interactions enabled by telematic performance’ (Ek, Östersjö, Visi and Petersson 2021: 2). Combining telepresence and copresence enables a focus on an indeterminate place – ‘the experience of place in tele-copresence as sometimes characterised by a mediated, liminal space’ (ibid.: 2). Within the project the ‘participants are compelled to develop new models of instrumentality for new modes of musicking’ (ibid.), and several settings involve live coding as a way of exploring hyperorgan affordances within a telematically mediated network.

In such settings, although all control messages are time-stamped and routed through a VPN server (Harlow et al. 2021: 6), making sure the latency is consistent and predictable, the lack of a physical, resonant body representing the remotely controlled instrument(s) creates a shared awareness of distance for the players, perceived as an indeterminate place where the musicking happens. Another important factor of the system presented here, which artistically explores latency, is the intricate connections between the performers in relation to the distribution of parameters of the live coding system. Those interconnected agencies create a shared embodied understanding of the sonic responsibility, where events played by any of the ensemble members can have a radical effect on the musical end result as they propagate through the system, altering parameters on their way.

This project is part of an ongoing inquiry of agency and polymorphism in musicking with live electronics, using live coding as a tool. Utilised as a musical instrument, live coding affords an acknowledgement of the characteristic flux and indeterminable ontological foundations of live electronics. Understood as a cybernetic musicking system that allows for multi-directional feedback and control flow between entities, it enables practices wherein all agents are part of a network of nodes, extending each other. Such a notion

has the potential to unlock an embodied understanding of a live electronic instrument as a set of algorithmically controlled human and non-human agents. Furthermore, combining the patch paradigm derived from modular synthesisers with the performative act of live coding opens up a shared parameter space and creative distribution of agency. This article discusses how performance within the new musicking context of the *Global Hyperorgan* project makes an exploration of specific affordances and a discovery of any unintended limitations of the system possible. Hereby, a collaborative adaptation process is enabled through which a shared instrumentality can emerge.

In the project, a new live coding environment called Paragraph was used to collaboratively control several geographically distributed hyperorgans over the internet (Harlow et al. 2021). The language was made as a syntactically minimalistic interface for SuperCollider's pattern library and was developed by the author, designed with modularity and distributed agency in mind. Modular systems in general and live coding in particular seem to fully take advantage of '[t]he unstable, fluctuating state significant for live electronic instruments, [...] as] one of their most defining characteristics' (Pettersson *in press*). The traditional division between designing an instrument and making use of its affordances – described by Nilsson (2011) as two parallel timelines labelled *design time* and *play time* – can easily merge into one, embodied performative action in such systems. Editing a line in a code block or switching patch cords around in a modular synthesiser 'can simultaneously build a new instrument and change the structure of the composition' (Pettersson *in press*). Foundational for the development of the Paragraph environment is an acknowledgement of the live coding musician's 'power to dissolve the traditional division of roles' (ibid.) of luthiers, composers and performers. A core feature of the system is the possibility of outsourcing certain musical gestures to other human and non-human agents, in order to unlock new combinations of a distributed embodiment of musical patterns. This involves possibilities to patch in control signals from a modular synth, various types of MIDI or OSC control or external audio. As such, the system combines the two paradigms of coding and patching and opens up for external agencies to interfere and play together with the live coder.

2. BACKGROUND

The act of live coding of music involves 'a meditative thought experiment or exercise in thought' (Cocker 2016: 103) and is situated at the intersection of composition, performance and instrument design. As an art form, it has matured from being a novelty

within the subculture of computer music into a well-established scene. This 'practice, operating at a critical interstice between different disciplines, oscillating between a problem-solving modality and a problematising, questioning, even obstacle-generating tendency' (ibid.: 107) has managed to keep a refreshing element of error and risk-taking in an otherwise more and more polished electronic music scene. Here, the performers 'actively disclose to an audience their moments of not knowing, of trial and error' (ibid.: 109). In further acknowledging uncertainty as an important performance quality, some systems for live coding allow for more or less intelligent agents to affect both the coder and the sonic structure or even to rewrite the code itself. The aesthetics and 'glorification of the typing interface' (Ward et al. 2004: 248) as mentioned in the TOPLAP Manifesto (ibid.: 247) is still prominent in the scene. However, the additional use of the laptop's own internal and external sensors and other controllers (see, e.g., Baalman 2016) have been promoted and exploited to the point where 'the performance of livecoding is influenced through its own side effects, transforming the code not only in the common logical, grammatical manner, but as an embodied interface of the machine on our lap' (Baalman 2020).

Another way of challenging the common live coding aesthetics is the use of traditional musical instruments as input devices, replacing the typing interface. One example is the Codeklavier project that 'aims to take advantage of the pianist's embodied way of making music and apply this to the act of coding' (Veinberg and Noriega 2018: 1). Replacing the performance interface of an instrument in such a way changes the instrumentality of live coding similar to how different controllers change the sonic affordances of any sound engine. Jeff Pressing uses a cybernetic analysis and the concept of 'dimensionality of control' (Pressing 1990: 12) to describe these affordances and the amount of expressive agency the musician has in the system. However, putting a live coding layer between the performance interface and the sound engine unlocks possibilities of dynamically changing these dimensionalities while playing. This in turn raises questions of how human and non-human agents interact within a system.

Playing with other live coders, distributing code and sharing the responsibility of an audiovisual result has been made possible with code libraries such as the Utopia (Wilson, Corey, Rohrhuber and de Campo 2022) and HyperDisCo (de Campo, Gola, Fraser, and Fuser 2022) extensions (Quarks) for SuperCollider. Another example is the web-based Estuary platform (Ogborn, Beverley, del Angel, Tsabary, and McLean 2017; Estuary 2022) that offers a collaborative, networked environment, supporting several different live coding languages for audiovisual experiments; for

example, explored by the SuperContinent ensemble ‘with an emphasis on developing Estuary’s utility for geographically distributed live coding ensembles’ (Knotts et al. 2021). Such web-based, telematic performance situations that ‘takes place without the acoustic and gestural referents of collocated performance scenarios’ (Mills 2019: 34) can aid in creating a sense of what Deniz Peters (2016) has called a shared or distributed instrumentality. He states that ‘whenever an instrument is played by multiple performers, and when, also, its bodily extension is multiple, then a compound sound or even single sound . . . might become the result of a joint intentionality’ (ibid.: 74). While Peters’s observations are made in a performance environment with musicians and instruments in the same room, Schroeder writes that ‘networked performative bodies are in a much greater space of suspense, of anticipated action, than performative bodies that share the same physical space’ (Schroeder 2013: 225). Many communicative habits that musicians make use of in traditional musicking (e.g., glances and breathing) are impossible in a telematic situation. Instead, ‘[n]etwork performance environments open up a rich space in which the performer needs to listen intently while being in a rather fragile, unstable environment’ (ibid.: 225). This fragility, and the lack of physicality, adds to the sense of an indeterminate place. To also take audiences into account and give them a chance to follow what is going on in such an environment can be challenging. Musicians might need to consider whether they want an audience at all, and if so, question whether they are listening *in* or *to* the indeterminate place. Further, one needs to discuss what agency they should have both in regard to their listening experience and in terms of understanding.

The work with the Paragraph environment, and its application within the TCP/IP Quartet and the *Global Hyperorgan* project, is situated in a wider context of computer music, algorithmic composition and improvisation. Akin both to George Lewis’s long-term work with computer driven co-players, acknowledging ‘the incorporation and welcoming of agency, social necessity, personality and difference as aspects of “sound”’ (Lewis 2000: 37), and the explorations of ‘meaning of intuition, musical structure and aesthetics by means of playing with a newly developed improvisation machine’ (Frisk 2020: 33) within the artistic research project *Goodbye Intuition* (Grydeland and Qvenild 2019), the Paragraph system itself aims to ‘represent the particular ideas of [its] creator’ (Lewis 2000: 33).

2.1. Intellectual effort on display

One of the core values of live coding, at least as it is practised today with the TOPLAP manifesto’s ‘show us your screens’ bullet (Ward et al. 2004: 247) still

resonating through the community, is to put the intellectual effort of musicking on display, usually by somehow projecting the code next to the performer. However, Palle Dahlstedt claims that if we are working with algorithms to make music and we want them to return good results, an embodied perspective is crucial (Dahlstedt 2018: 42). He argues for different strategies where algorithms can be anchored in the physical world and where they become possible to ‘relate to, play with, and react to . . . in very much the same way as we relate to music coming from fellow musicians’ (ibid.: 44). One of the examples is the exPressure Pad, an instrument that explores dynamic mapping strategies (see, e.g., Dahlstedt 2008), studied as a performer together with Per Anders Nilsson in their duo pantoMorf. Utilising pressure sensor-based controllers, the instrument explores a complex parameter space of timbral trajectories, but keeps some of the basic designs usually found in acoustic instruments such as the volume being proportional to physical effort and that ‘every change in sound corresponds to a physical gesture’ (Dahlstedt 2018: 46). Similar ideas have also been implemented as hyper-instrumental extensions to keyboard instruments, enabling ‘an intricate search algorithm through . . . physical playing, in an embodied exploration which at the same time is music making’ (ibid.: 47). The notion of an embodied search algorithm explored in front of an audience has obvious connections, not only with free-improvised music in general, but also with the tradition of projecting your algorithms during a live coding performance, avoiding all kinds of obfuscation, putting yourself at risk and simulate a state of emergency:

The use of algorithms allows for the breaking of habits through the suggestion of new combinations and timings of movement, allows for the generation of new material and also provides a space for playing with improvisational movement ideas. (Sicchio 2021)

In computer music we do not need to care about a correlation between effort and result. The decision to interconnect them is up to the composer or performer. On the opposite end of the continuum where a traditional live coding performance, with code and intellectual effort projected on the wall is on one end, we find the concept of acousmatic listening. Here, the act of non-referential listening is put in focus by avoiding visual stimuli and, when necessary, trusting the sound itself to carry the understanding of a composition. However, as Godøy (2006) points out, listening is a multi-modal activity, and we understand sonorous objects through our bodies. He further states that ‘[t]he idea of motormimetic cognition implies that there is a mental simulation of sound-producing gestures going on when we perceive and/or imagine music’ (ibid.: 155), which suggests that the idea of the

acousmatic might actually be impossible. From a wider, musicological perspective, Christopher Small argues that this extreme focus on listening within Western art music is a special type of musicking, which, to a large extent, is driven by capitalism (Small 1998). He exemplifies with the complex economics associated with traditional concert halls and further explains how these ‘place[s] for hearing’ (ibid.:19) separate themselves from the outside world to enable a ritualistic act of listening. According to Small, these, along with the record industry, have had such a strong influence on the Western idea of music that we have almost forgotten that the main purpose can also be the actual making of it, that is, the performative act could just as well be the principal aim and more important than the sonic result.

Dahlstedt seemingly puts a strong emphasis on effort-based instruments to enable empathetic perception by others and possibly a sense of understanding of underlying complex algorithms by an audience. Such instruments allow for traditional synchronisation if ‘the presence of performers and listeners who physically share the same space provides the framework for synchronization’ (Chagas 2006, quoted in Dahlstedt 2018: 53). If ‘[m]usical embodiment is a temporal experience that requires the synchronization of temporal objects and events’ (ibid.), then such instruments might be less suitable for, for example, telematic performances where the lack of physical instrumental bodies and spaces are obvious. However, ‘algorithms can, in several different ways, help create an augmented mind’ (ibid.: 57). They can aid in a performance, as a tool for memorising or open up for non-linear thinking. Here, the algorithm becomes part of a cybernetic feedback loop, where the human and the machine (the algorithm) inform each other.

In the modular synth community there is a growing number of both professional and amateur musicians that use social networks for a kind of introspective musicking, akin to the TOPLAP manifesto’s call for ‘show[ing] us your screens’ (Ward et al. 2004: 247). Here, instead of code, the idea is to share patch experiences or so called *noodles*, not necessarily intended for pure listening enjoyment. The focus lies on the performative act of patching and musicking with modular systems. An interesting aspect of this is that there seems to exist a common, quietly negotiated agreement on the high value of these noodles. The shared experience of understanding and when the act of patching becomes an extension of your musical mind appears to be central here. There are of course critics looking for a pure listening experience, which, as Christopher Small puts it, takes for granted that music is intended for ‘listening to rather than performing, and ... that public music making is the sphere of the professionals’ (Small 1998: 71), but the

question of how it sounds seems secondary here. Similar situations, where the sonic material’s social function is primary, for example, exist within folk music and in rave culture, where dancing and interacting with other humans can be more important than the listening experience. A party night experience at a techno club where the DJs adapt their sets to the vibe on the dance floor is another example of such shared agencies.

2.2. Modularity and shared agency

A significant feature of a modular system is its ability to change. Thus, a modular synthesiser is more of a set of musical intentions than an instrument. It is a collection of ideas for sound design and musical structures, divided into subsets of functions, patched together for complex results. In such a system it is the patch that becomes an instrument or a composition, rather than the modules the system comprises, and ‘[t]he unpatched modular synth is thus nothing more than a possible future actualisation’ (Pettersson *in press*). On the other hand, the patched-up modular system – with all its components and agencies – becomes an embodiment of intentions. It is also a system in flux where a live performance can involve everything from slight adjustments of single parameters to a transformational re-patching into a new instrument. Accordingly, Magnusson and McLean refer to the patch programming interfaces of these instruments as a special kind of tangible live coding (Magnusson and McLean 2018: 250).

Cybernetic theories of feedback and control have also been thoroughly discussed and explored within modular synth communities. One example is the thread ‘Cybernetics and AI with Serge’ (Haslam 2020a), started by the musician Gunnar Haslam’s alias mfaraday on the Modwiggler forum. Using a small modular synth, he shows ways to explore concepts such as neurons and neural networks for artistic purposes on his YouTube channel *La Synthèse Humaine* (Haslam 2020b). Akin to Dahlstedt (2018), Haslam also takes a stand against art produced using these algorithms without human input, referring to classic cybernetic literature by Norbert Wiener (e.g., Wiener 1948, 1956) but also socialism and politics as in the writings of Karl Marx. For him, the goal is to explore a cybernetic system, rather than to create something predictable, and human agency is crucial as input to the systems in all these cases, asking the listener ‘to use [their] humanity to guide [them]’ (Haslam 2020c). The focus in such systems lies on how ‘all the actors co-evolve’ (Latour 1991: 117) and how ‘[a] cybernetic ontology would refuse any strict division between the human and non-human’ (De Souza 2018: 159). The sociological aspects of musical

instruments are also interesting in regards to accessibility. Live coding environments are usually free and open source software and within the modular synth community there is a strong tendency towards DIY. This opens up for practically anyone with a computer and time and inclination to engage with the scene. Other musical instruments, especially venue dependant ones such as pipe organs, are traditionally more inaccessible, but technology can aid in diversifying access to such instruments.

2.3. Hyperorgans

In general, hyperinstruments are referred to as traditional instruments, extended by means of technology. The reasons are often to explore new instrumental affordances and uncharted sonic territories, but these extensions can also be added to provide ‘unprecedented creative power to the musical amateur’ (Machover 1992: 3). Both reasons are relevant in regard to large pipe organs. As instruments surrounded with strong musical traditions and even religious symbolism, they are generally difficult to access, not only physically, but also in regard to gaining the necessary knowledge and trust to work with them. By extending instruments with such hyper-functionality, artists and musicians from other fields and backgrounds can gain access and at the same time develop novel approaches to their performance.

Similar to a modular synthesiser, an organ could also be thought of as a collection of subsystems that comprise a set of musical intentions:

The registers manifest fixed additive synthesis in which different oscillators are blended to create complex periodic waveforms. Selecting stops and directing wind to different sets of pipes is analogous to distributing control voltages around by means of patch cords in a modular system. Additionally, organs often include different kinds of mechanical filters and modulators such as wooden shutters and tremulants. (Harlow et al. 2021: 4)

Thus, the different musical functions of an organ could already be thought of as timbral extensions to one another. In a hyperorgan, such as the Utopa Baroque organ (Fidom 2020) at Orgelpark in Amsterdam, the signal flow has been made externally accessible through control interfaces. By means of protocols such as OSC and MIDI, this unlocks new possibilities for interaction and enables ‘[t]he inner modularity of the hyperorgan [to] be expanded with new modules, human and non-human, and even other hyperorgans’ (Harlow et al. 2021: 5). Furthermore, it opens up for ontological questions regarding both material and immaterial aspects of the organ as a musical instrument. Bringing in telematic performance and live coding into the mix also enables sociological

perspectives of access to these instruments and to the spaces where they are usually situated.

2.4. Musicking with patterns

Already in the early 1980s Laurie Spiegel (1981) made an attempt to provide a generalised set of useful pattern operations in music intended for computer musicians. As patterns in programming languages are often represented as lists of numbers, they can easily be manipulated in all the ways that she suggests, including, for example, transposition, reversal and rotation operations. In most languages there are extensive possibilities for list manipulation, but to make them correspond to intended musical results, aesthetic choices need to be made. As shown by Magnusson and McLean (2018), even a seemingly simple and straightforward operation, such as reversal, requires many interpretative artistic and aesthetic choices. Thus, instead of trying to unify methods in tools for musicking, Magnusson and McLean argue for systems where the ‘method names ... become semantic entities in the compositional thinking of the composer or performer [that] outline the scope of the possible’ (ibid.: 262). Such semantic entities, with the ability to permeate and transform a musical mindset, could also be found in classic composition techniques such as counterpoint. For example, the notion of a *cantus firmus*, a foundational melodic line from which other material is derived by means of pattern operations similar to those listed by Spiegel (1981), could be understood as algorithmic composition carried out within, and taking advantage of the realm of traditional staff notation. In acknowledging that the music we write is influenced by such notation and how we leave important parts of the music to the interpreter to add into that system, Magnusson and McLean also discuss how programming languages can aid in going ‘beyond the usual dimensions of music notation’ and instead become ‘a process of live exploration, rather than description’ (Magnusson and McLean 2018: 246). However, programming languages can also be ‘designed with increasing constraints, to the degree that they can be seen as musical pieces in themselves’ (Magnusson 2013: 1). In such a system a composer’s artistic manifest could be inscribed into the language, not only similar to how a musical instrument embodies music theory (Magnusson 2019), but also in how it proclaims a strong work identity.

3. PARAGRAPH

3.1. The live coding system

Paragraph is a custom-made live coding interface for the creation of and interaction with SuperCollider

patterns, which includes modular signal routing facilities, discrete as well as ambisonics panning, and takes advantage of multi-core processors (utilising the Supernova server). It stems from artistic practice and personal compositional techniques developed by the author over many years and was made as a tool to formalise and simplify coding of such patterns, and at the same time be able to interface with external signals in an efficient way. The name comes from the idea of writing short paragraphs that are representations of musical ideas, with the notion that everything written within the same paragraph technically refers to the same receiver (a pattern), and artistically to the same musical idea. The section sign, §, (which is called *paragraf* in Swedish) is used to syntactically refer to receivers. It was chosen partly due to the translational pun, and partly due to its accessibility, just below the escape key, on a Swedish keyboard layout. Receivers can be either patterns, where the section sign is followed by a number or unique name (e.g., §1 or §kick), or refer to the whole system with only the § symbol.

The syntax could be thought of as a dialect, derived from the standard SuperCollider syntax, constructed using the built-in preprocessor to filter the input code and return valid syntax. A basic objective was to keep the code interface minimal with as few delimiters as possible, in order both to fit on small screens and to be fast and responsive. Thus, spaces and line breaks are the only separators used, and the most common keywords have been abbreviated to either one or three letters. For example, the regular SuperCollider Event keys `\instrument`, `\degree`, `\dur` and `\legato` have been replaced by `i`, `n`, `d` and `l`, respectively. Value patterns that set those keys are expressed in a similar abbreviated way, (e.g., `wht 0.1 0.5`) instead of SuperCollider's regular syntax `Pwhite(0.1, 0.5)`. Many useful default values for patterns and devices for generating or processing sound are predefined in the backend of the system. In SuperCollider lingo, this, for example, includes a default chromatic scale, a number of SynthDefs for playing back samples and simple synths for routing signals in the system, but the library is easily extendable as new needs and ideas arise. Figure 1 is a simple example of a Paragraph pattern named §1 playing such a predefined synth by setting the instrument parameter `i` to `\sine` on line 1. On line 2, the note degrees, `n`, are set by a `rnd` pattern that randomly picks a value from an array. In this example, the default chromatic scale is used (i.e., not set explicitly) and single note values and chords, expressed as sub arrays, are combined. A chromatic transposition value, `exp`, is added to the note degrees on line 3, also chosen by means of a `rnd` pattern, but here with a supplied second array of probability weights. Both the duration `d` and legato `l` parameters on lines 4 and 5

```

1 §1 i \sine
2 n rnd [0 [1 4 7] -3 6 [5 9]]
3 exp rnd [0 -12 12] [0.8 0.1 0.1]
4 d wht 0.1 0.5
5 l wht 0.1 0.3
6 div rnd [1 2 3 5 8]
7 prb 0.4
8 a seq [0.5 0.3 (rnd [0 0.2 0.1] 4) 0.05] | \sin 0.4 0.2
9 atk 0.01
10 rel 0.9
11 sat seq [0.2 0.1 0.0] !(seq [5 4 3 2 1])

```

Figure 1. Example of basic Paragraph syntax.

are set by another type of random generator – the previously mentioned `wht` – which produce uniformly distributed values within the given range. The `div` parameter on line 6 subdivides and repeats the current event's duration and the `prb` value of 0.4 sets a probability for an event to be played to 40 per cent. For the amplitude `a`, set on line 8, a sequence (`seq`) of numbers is used. The array of sequenced values also shows an example of a nested pattern, enclosed in parentheses. Here, the sequence enters a random pattern that chooses between three values four times before moving on. Further, the values produced are modulated by an independently running sine wave LFO with a frequency of 0.4 Hz and an amplitude of 0.2. As shown here, modulators are expressed with the `|` symbol and, besides simple oscillators, they can also be user-defined functions, hardware, OSC or MIDI inputs. All sounds generated by or passed through the pattern system can be processed by means of a global amplitude envelope, low and high pass filters, a tempo synced delay, dynamics and saturation. In Figure 1, the amplitude envelope and the saturation is set. On lines 9 and 10, the envelope's attack and release times have been set to fixed values. Finally, on line 11, the saturation parameter (`sat`) is sequenced by a pattern that gets its values duplicated a number of times according to another sequence. Here, Paragraph uses the commonly used exclamation mark shortcut for duplication of values, found in standard SuperCollider syntax.

The system is modular and can integrate external signals, hardware and controllers as well as internal modules (e.g., instruments, effects, midi devices, VST plugins, cv). From a user perspective, those modules are syntactically similar and, once defined in the system, referred to by a chosen name. The `def` method handles these definitions, following the syntax `[def type nameTag arguments]`, where the number and type of arguments vary depending on the type. Reading from the top in Figure 2, line 1 defines a name tag for a sample and loads it into a buffer. Here, the provided file path points directly to a sound file. A folder of samples can also be defined in a similar way as shown on line 2, here labelled with the tag `\kit`. All samples in that folder now become

```

1 § def snd \bd "/path/to/sample/909samples/909BD02.wav"
2 § def snd \kit "/path/to/folder/808samples/"
3 § def mid \esq1 "mioXL" "DIN 4" 6
4 § def efx \vrb \reverb
5 § def efx \wrp \autoWarp
6
7 §2 i seq [\bd \kit0 kit9 \esq1 \kit13]
8 n seq [-12 3 0 (rnd [[0 3 7] [2 4 6] [-1 2 5]]) 12]
9 vrb 0.2
10 wrp rnd [0 0.1] [0.9 0.1]

```

Figure 2. Defining and playing instruments and effects in Paragraph syntax.

accessible as instruments from any pattern as `\kit` followed by a number referring to the index of the sample in that folder. It is also possible to assign an array of several tags (e.g., `[\kit \drums \perc]`) to the same sample or folder of samples in order to facilitate different organisation of larger sound banks. An external MIDI instrument is defined on line 3. Here, besides the unique name tag, we also need to supply the MIDI device and port as well as the physical hardware input that the instrument is connected to. On lines 4 and 5 two send effects are defined. As mentioned earlier, the backend of the Paragraph system includes a number of such effects, written as SynthDefs in the regular SuperCollider syntax. Those are defined for use in patterns by providing a name tag and the name of the SuperCollider SynthDef to use. Figure 2 shows how these send effects can be used in a pattern (lines 9 and 10).

Departing from my own practice, tools and aesthetics, Paragraph also facilitates musical ideas and concepts often returned to, to be expressed in an efficient and comprehensible way. As such it implements a personalised composer's tool kit that includes certain methods for pattern generation and manipulation, developed and refined to fit my own musical voice. It is beyond the scope of this article to go into all the details of this toolkit, but the previously discussed notion of a *cantus firmus* has, for example, been implemented into the language. These are also patterns, but they run independently on a global hierarchical level. Thus, the data output of them could be shared among patterns on the lower level. A *cantus firmus* pattern consists of two subpatterns – one for the values and one for the durations of those values. Single or several such *cantus firmus* patterns could be defined, thus enabling the notion of co-players with agency over other pattern algorithms. In Figure 3, such a *cantus firmus* pattern with the name `\chords` is defined on line 1. This is used to set the scale (`scl`) in three patterns that independently play over the note degrees in these chords. The score next to the code shows one possible result of four bars, rendered as traditional notation. However, due to the randomness in the patterns each new rendition will be different.

3.2. The TCP/Indeterminate Place Paragraph system

The TCP/Indeterminate Place Quartet (Figure 4) was formed with an intention to explore different types of interactions with the hyperorgans in the context of the *Global Hyperorgan* (Harlow et al. 2021) project. A modular approach was used, both to the organs and to the instruments and agencies of the quartet. The notion of an organ as a modular system enables the addition of both new controllers and sonic extensions. Here, these extensions are other musical instruments, involving Robert Ek's clarinet fitted with a motion sensor on the bell (ibid.), an electric MIDI guitar played by Stefan Östersjö, and Federico Visi performing with a laptop and gestural controllers. Visi's setup navigates a sonic corpus of pre-recorded guitar sounds that is sonically extending Östersjö's guitar and sending controller data to the organ network. Finally, the Paragraph live coding system described earlier, functions as an instrument played by the author of this text. Thus, the auditory output of a performance consists of the controlled local and remote organs, the clarinet, the electric guitar and any sounds produced or processed by the laptops. Visually, the organs have been a dominating part of the audiovisual stream due to the nature of the project, but also code projected on or next to organs have occurred.

From the perspective of the live coder, the Paragraph system, as it is used within the TCP/Indeterminate Place Quartet, can be understood as a complex arpeggiator playing geographically distributed instruments. So far, these instruments have involved different types of pipe organs, extended with remote control facilities such as MIDI or OSC. In a regular arpeggiator, the pitches corresponding to keys pressed on a keyboard are played one by one according to a selected pattern and synchronised to a clock. In many synthesisers featuring arpeggiators, patterns can be changed on the fly, and continuous modulation sources such as after touch, wheels and joysticks can be mapped to articulation (i.e., envelope parameters) and timbre qualities. This allows for less static performance. Staying within the analogy of arpeggiators, in this setup (Figure 5), the function of pressed keys is outsourced to a MIDI guitar. The guitar sends note numbers on separate MIDI channels, one for each of the six strings. These channel numbers are used to index the note values and store them in an array of six elements in the Paragraph system (Figure 5). In practice, this can be thought of as dynamically altering the pressed keys for the generated note events. Similar to the *cantus firmus* example earlier (Figure 3), the live coder maps the guitar-controlled array to the scale parameter in the patterns. Unlike a common arpeggiator, the patterns playing the chord notes are not limited to simple directional or

```

1 § def cf \chords (seq [[0 3 5] [2 4 7] [-1 2 4]]) (rnd [8 12 16])
2
3 §1 i \saw
4 scl \chords
5 n rnd [0 1 2]
6 d seq [0.25 0.5 0.75]
7
8 §2 i \saw
9 scl \chords
10 n wht 0 5
11 cxp 12
12 d 1/3
13
14 §3 i \saw
15 scl \chords
16 n [0 1 2]
17 d 4

```

Figure 3. A *cantus firmus*-driven pattern in Paragraph and one possible outcome in standard notation.



Figure 4. The *TCP/Indeterminate Place Quartet* performing as part of the NIME 2021 conference, controlling the Skandia Cinema Organ in R1 (pictured), Stockholm, and the Utopa Baroque Organ in Orgelpark, Amsterdam, 16 June 2021.

random ones. Instead, those patterns can be of arbitrary length and produce both note degrees and chords within this scale, taking advantage of both SuperCollider's extensive pattern library and any user-defined pattern algorithms. Typically, scales are represented as ascending arrays of numbers; for example, [0, 2, 3, 5, 7, 8, 11] for a harmonic minor scale. Scrambling such an array to something like [5, 3, 2, 7, 0, 11, 8], means that a note degree of 0 would not return the tonic (0) as one would expect, but instead 5. A degree of 6 returns 8 instead of 11 and so on. From a

music theory perspective, such unsorted arrays are rather considered as sets of pitches than proper scales. While the initial idea indeed was to sort the incoming guitar notes incrementally, the affordances of an unsorted array, which instead independently stored the last played note of the respective guitar string, turned out to be more musically rewarding. The history of played notes is projected as sound events in the system of pipe organs through the live coded pattern, thus shaping the context in which the future notes are played by the guitarist. This particular

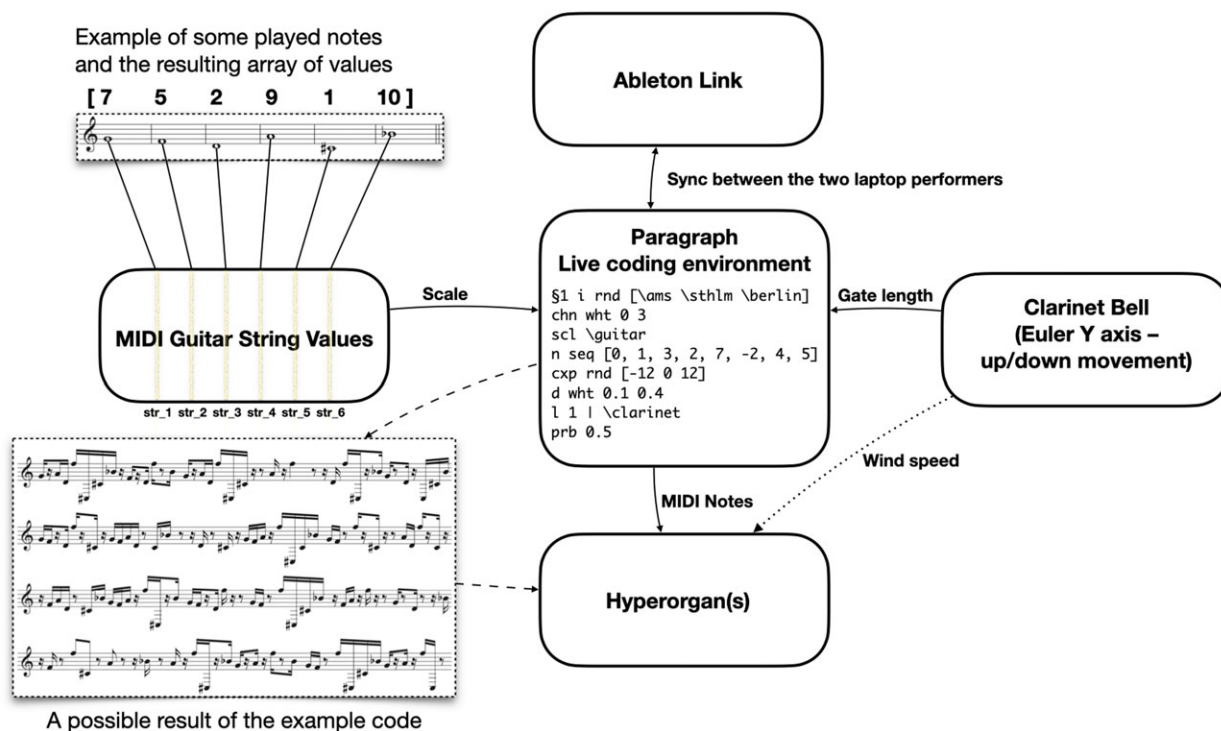


Figure 5. The *TCP/Indeterminate Place* system from the perspective of the live coder.

affordance of the system unlocks both the notion of mimicry and opposites – where, for example, the guitarist can mimic the pattern generated by the live coder while at the same time rewriting the scale utilising the guitar as a live coding interface, and the coder can in turn decide to either follow or go against the texture produced by the guitarist.

Further, the gate length of the note events (i.e., the legato parameter, *l*) produced in Paragraph have been outsourced to the clarinetist, playing a customised instrument with a sensor-fitted bell, sending data wirelessly to the system. In this setup, the up and down movement (the Euler *y*-axis (pitch) of the sensor) of the clarinet was used to dynamically alter the note events produced between staccato and legato. The clarinet bell was also used to directly control the wind speed of the organ, which enabled new sonic possibilities beyond the tonal. Similar to the agency of the guitarist, the connection between the Paragraph system and the clarinetist establishes a multi-directional system of shared instrumentality where the sonic affordances of the live coder's patterns can be radically altered by the clarinetist, who in turn acts within the dynamically changing performance ecosystem (Waters 2007) created by the quartet.

As a final outsourced parameter, the control of tempo was shared between the live coder and the second laptop performer (Federico Visi) in the quartet. This was accomplished by means of Ableton Link

(Golz 2018). Utilising a Max patch and gestural controllers to navigate both a sonic corpus of pre-recorded guitar material and to send note and controller data to the connected organs, this performer's instrument also incorporates sequencing and looping (see Harlow et al. 2021 for further description). In practice, dynamic global tempo changes were usually avoided during the performance. Instead, clock dividers and multipliers were used to individually relate to the common clock.

The distributed parameter space utilised in this system thus forms a shared *cantus firmus*, exerting an obvious agency on the live coder. Controlled by the three other players, it forces an exploration of Paragraph's affordances as a constrained instrument, where some of the most important parameters are outsourced. Further, as an aesthetic decision was made not to use any other sounds or treatments, except the controlled hyperorgan, the system limits the live coder to a pure exercise in quick adaptations to the incoming data, thus enabling an in-depth exploration of pattern affordances.

A similar system has been used in several different performances, both with the full quartet, sometimes physically dispersed, performing on one or more hyperorgans, and in duo settings with live coder and guitar or live coder and clarinet. With the full quartet, we have also performed on geographically distributed organs in Piteå and Amsterdam (see Video Example 1

and Visi 2021a), and Stockholm and Amsterdam (see Video Example 2 and Visi, 2021b), but also as part of larger constellations of musicians, telematically playing five hyperorgans located in Amsterdam, Düsseldorf, Piteå, Stockholm and Berlin.

4. DISCUSSION

In a typical telematic performance situation, the lack of a physical resonant body and the missing natural blend of timbres that occurs in a real room, presents us with a very particular set of affordances. Sometimes the indeterminate place for musicking created by this situation can be an overwhelming force that needs careful consideration. A possible strategy is to deal with this agency as a co-player, as much a part of the tele-copresence (Ek et al. 2021) as the human agents in the system. Factors such as long or varying latency and bad auditory feedback can then be viewed as a musical voice of one of the agents, instead of simply failing technology. This notion can also aid in creating an embodied understanding of the indeterminate place because it unlocks a musicking context where, for example, live coded sonic events are gesturally shaped by such technical aspects. In combination with multi-modal listening to those ‘gestural sonorous objects’ (Godøy 2006), the telematic situation can become comprehensible. From the player’s perspective, the indeterminate place then becomes a virtual resonant body, and the algorithms and instruments exciting it become multi-directional extensions to both agencies. To communicate this perceived embodiment and intellectual effort of musicking within such a complex system, it seems reasonable to use a multi-modal representation. In our performances with the TCP/Indeterminate Place Quartet we have so far used several cameras and microphones at each venue hosting the connected hyperorgans. Sometimes the microphones have been placed inside the organ in order to minimise the impact of the local room acoustics. The audiovisual streams are then mixed and cut live by an external technician, before broadcasting the stream to an audience. The sense of an indeterminate place, which is a result of musicians and instruments being in different physical locations, is accentuated by the mixing process and plays an important part for the experience of this new telematic room. In this article, the focus has been on the live coders perspective from within an interactive system. A thorough study of the implications of spatialisation and mixing are thus beyond the scope of this text, but it should be acknowledged that the aesthetics of the presentation and the consequences of these factors both from a performer and a listener perspective are of great importance. In practice, these performance settings entail that the sounding outcomes may take

radically different shapes in different ends of the system, generating quite different experiences for performers in the quartet and a remote audience. In some cases when the quartet have all been gathered in the same venue, next to one of the connected organs, with a strong sense of playing well together in the same acoustic space, the audience watching the stream instead heard glitches due to sudden lack of bandwidth or failing connections. On these occasions, the setting of the telematic performance itself brutally exposes the nature of musicking as an act of the present, and that music is an articulation of time. Further, the often variable latency of a telematic performance situation can make it difficult to maintain a sense of playing together. As mentioned in the introduction, latency is already an artistically defining factor of any live coding. While the other musicians in the quartet only deal with the latency of the telematic system, the live coder thus experiences a double latency. In response, the live coder can either try to predict how the others might play in the future and adjust the code accordingly, or simply trust that the others will adjust to any presented material. Both cases are problematic in regard to artistic equality within an ensemble that strives for non-hierarchical musicking situations. The modular system and interconnections between the members of the quartet compensates to some extent for these issues. In order to enhance the multi-modality of the system for the future, we have discussed giving the listener agency over which parts of the stream to focus on; for example, by means of a smartphone app. Other possibilities considered include extensions to the Paragraph environment that allow for (a meta level of) control of presentation parameters from the musician’s side.

In *Terpsicode*, a somewhat related project in regard to shared agency, Kate Sicchio (2021) has created a live coding language for creating visual scores for dancers. By ‘refer[ing] to the computational systems and approaches and not the specific medium or outcome of the systems’ (ibid.), it is possible to utilise these tools in novel ways. Sicchio’s language is built on dance terminology, akin to Magnusson and McLean’s argument for ‘semantic entities in the compositional thinking ... [that] outline the scope of the possible’ (Magnusson and McLean 2018: 262). When Sicchio discusses the importance of sharing agency from the perspective of a choreographer, she states that ‘not just the choreographer but the dancers too should be afforded the privilege of expressing their own creative—human—thoughts. They are not a machine but are simply responding to a machine, much like a live coder’ (Sicchio 2021). However, a division of human and machine might conflict with a cybernetic understanding, as ‘a cybernetic epistemology would claim that the system as a whole—not just the human within it—

thinks, improvises, and so forth' De Souza (2018: 10). Thus, viewed as a cybernetic system, our humanity becomes only one of many agencies.

In Paragraph, the aim has been to build my own compositional aesthetics and strategies into the language in order to simplify its practice and to enable real-time composition by means of, what Cocker refers to as, 'the meletē of live coding' (Cocker 2016: 102). Another aim has been to incorporate it into a network of agencies. Similar to Magnusson's idea of a live coding system so constrained that it could be considered as a piece in itself (Magnusson 2013) but also to some extent to his minimal pattern language *ixi lang* (Magnusson 2011), Paragraph could be considered a piece rewritten over and over again. As previously mentioned, it incorporates methods for defining *cantus firmi*, foundational patterns that can pipe data into other (sub)patterns, but metaphorically, the system itself could also be considered a personal *cantus firmus*, permeated by a subjective idea of musicking and compositional practice. Similar to George Lewis's description of *Voyager* (Lewis 2000), Paragraph is a 'composing machine that allows outside intervention' and also functions as an 'improvising machine ... that incorporates dialogic imagination' (Lewis 2000: 38), emanating from the foundational question of how my own compositional practice has co-evolved with SuperCollider as a tool.

The parallel timelines of designing and playing merge in interesting ways when developing your own live coding environment. The system is constantly changing as new compositional or performative needs arise. To add new functions and methods often leads to a better understanding of the things you tend to return to as a composer-performer. On the other hand, it can also hide complex algorithms that sabotage the notion of putting intellectual effort on display. It can obstruct the sense of liveness and the possibility of giving the audience an opportunity to follow your thinking-in-action (Schön 1983; Cocker 2016). In relation to the *from scratch* coding practice, where performers start from a blank page in a language often designed by someone else, it could be argued that one could never start from scratch in a self-made system if many shortcuts to reusable code are implemented as method calls. The Paragraph environment constrains certain coding behaviour and promotes others, and these shortcuts could definitely be criticised from the perspective of obscurantism. However, these very features also enhance the sense of work identity being incorporated within the language.

SUPPLEMENTARY MATERIAL

To view supplementary material for this article, please visit <https://doi.org/10.1017/S1355771823000377>

REFERENCES

- Baalman, M. A. J. 2016. Interplay between Composition, Instrument Design and Performance. In T. Bovermann, A. de Campo, H. Egermann, S-I. Hardjowirogo and S. Weinzierl (eds.) *Musical Instruments in the 21st Century: Identities, Configurations, Practices*. Singapore: Springer, 225–41. https://doi.org/10.1007/978-981-10-2951-6_15.
- Baalman, M. 2020. Code-livecode-live. <https://marijebaalman.eu/projects/code-livecode-live.html> (accessed 15 September 2022).
- Cocker, E. 2016. Performing Thinking in Action: The Meletē of Live Coding. *International Journal of Performance Arts and Digital Media* 12(2): 102–16. <https://doi.org/10.1080/14794713.2016.1227597>.
- Dahlstedt, P. 2008. Dynamic Mapping Strategies for Expressive Synthesis Performance and Improvisation. *Computer Music Modeling and Retrieval. Genesis of Meaning in Sound and Music*, 5th International Symposium, CMMR 2008, Copenhagen, Denmark, May, 227–42. https://doi.org/10.1007/978-3-642-02518-1_16.
- Dahlstedt, P. 2018. Action and Perception: Embodying Algorithms and the Extended Mind. In A. McLean and R. T. Dean (eds.) *The Oxford Handbook of Algorithmic Music*. Oxford: Oxford University Press, 41–65. <https://doi.org/10.1093/oxfordhb/9780190226992.001.0001>.
- De Campo, A., Gola, B., Fraser, G. and Fuser, G. 2022. HyperDisCo. <https://github.com/aiberlin/HyperDisCo>.
- De Souza, J. 2018. Orchestra Machines, Old and New. *Organised Sound* 23(2): 156–66. <https://doi.org/10.1017/S1355771818000031>.
- Ek, R., Östersjö, S., Visi, F. G. and Petersson, M. 2021. The TCP/IndeterminatePlace Quartet: a Global Hyperorgan Scenario. *International Conference on New Interfaces for Musical Expression (NIME)*, Shanghai, China.
- Estuary. 2022. estuary.mcmaster.ca.
- Fidom, H. 2020. *The Utopa Baroque Organ at the Orgelpark*. Orgelpark Research Reports vol. 5, part 2. www.orgelpark.nl/orr/5/2/web/ (accessed 15 September 2022).
- Frisk, H. 2020. Aesthetics, Interaction and Machine Improvisation. *Organised Sound* 25(1): 33–40. <https://doi.org/10.1017/S135577181900044X>.
- Godøy, R.-I. 2006. Gestural-Sonorous Objects: Embodied Extensions of Schaeffer's Conceptual Apparatus. *Organised Sound* 11(2): 149–57. <https://doi.org/10.1017/S1355771806001439>.
- Golz, F. 2018. Ableton Link – A Technology to Synchronize Music Software. *Proceeding of the LinuxAudio Conference 2018*, 39–42. <https://doi.org/10.14279/depositonce-7046>.
- Grydeland, I. and Qvenild, M. 2019. Goodbye Intuition. www.researchcatalogue.net/view/974962/974963 (accessed 15 January 2022).
- Harlow, R., Petersson, M., Ek, R., Visi, F. and Östersjö, S. 2021. Global Hyperorgan: A Platform for Telematic Musicking and Research. *International Conference on New Interfaces for Musical Expression (NIME)*, Shanghai, China. <https://doi.org/10.21428/92fbeb44.d4146b2d>.
- Haslam, G. [mfaraday]. 2020a. Cybernetics and AI with Serge [online forum post]. *Modwiggler*. www.modwiggler.com/forum/viewtopic.php?t=237984 (accessed 15 September 2022).

- Knotts, S., Betancur, C., Khoparzi, A., Laubscher, M., Marie, M., Ogborn, D. et al. 2021. Negotiating Shared Live Coding Practices in SuperContinent, an Online Laptop Ensemble. *eContact!* 21(1). https://econtact.ca/21_1/knotts-etal_supercontinent.html (accessed 15 September 2022).
- Latour, B. 1991. Technology is Society Made Durable. In J. Law (ed.) *A Sociology of Monsters: Essays on Power, Technology, and Domination*. London and New York: Routledge, 103–31.
- Lewis, G. E. 2000. Too Many Notes: Complexity and Culture in Voyager. *Leonardo Music Journal* 10: 33–9.
- Nilsson, P. A. 2011. A Field of Possibilities: Designing and Playing Digital Musical Instruments. PhD thesis, University of Gothenburg.
- Machover, T. 1992. *Hyperinstruments: A Progress Report, 1987–1991*. Cambridge, MA: MIT Media Laboratory. https://opera.media.mit.edu/publications/machover_hyperinstruments_progress_report.pdf (accessed 18 July 2023).
- Magnusson, T. 2011. Ixi lang: A SuperCollider Parasite for Live Coding. *Proceedings of ICMC 2011*, Huddersfield, UK.
- Magnusson, T. 2013. The Threnoscope: A Musical Work for Live Coding Performance. *First International Workshop on Live Programming in conjunction with ICSE 2013*, San Francisco, CA, USA.
- Magnusson, T. 2019. *Sonic Writing: Technologies of Material, Symbolic, and Signal Inscriptions*. New York: Bloomsbury Academic.
- Magnusson, T. and McLean, A. 2018. Performing with Patterns of Time. In R. T. Dean and A. McLean (eds.) *The Oxford Handbook of Algorithmic Music*. Oxford: Oxford University Press, 245–66. <https://doi.org/10.1093/oxfordhb/9780190226992.013.21>.
- Mills, R. 2019. *Tele-Improvisation: Intercultural Interaction in the Online Global Music Jam Session*. Berlin: Springer.
- Ogborn, D., Beverley, J., del Angel, L. N., Tsabary, E. And McLean, A. 2017. Estuary: Browser-Based Collaborative Projective Live Coding of Musical Patterns. *International Conference on Live Coding (ICLC)*, Morelia, Mexico.
- Peters, D. 2016. Instrumentality as Distributed, Interpersonal, and Self-Agential: Aesthetic Implications of an Instrumental Assemblage and Its Fortuitous Voice. In T. Bovermann, A. de Campo, H. Egermann, S-I. Hardjowirogo and S. Weinzierl (eds.) *Musical Instruments in the 21st Century: Identities, Configurations, Practices*. Singapore: Springer, 67–78. https://doi.org/10.1007/978-981-10-2951-6_15.
- Petersson, M. In press. *A New Morphology: Strategies for Innovation in Live Electronics Performance*. Stockholm: Royal College of Music.
- Pressing, J. 1990. Cybernetic Issues in Interactive Performance Systems. *Computer Music Journal* 14(1), 12–25.
- Schön, D. A. 1983. *The Reflective Practitioner: How Professionals Think in Action*. New York: Basic Books.
- Schroeder, F. 2013. Network[ed] Listening: Towards a De-centering of Beings. *Contemporary Music Review* 32(2–3): 215–29.
- Sicchio, K. 2021. Live Coding Choreography with Terpsicode – Enabling agency for the human interpreter. *eContact!* 21(1). https://econtact.ca/21_1/sicchio_terpsicode.html (accessed 15 September 2022).
- Small, C. 1998. *Musicking: The Meanings of Performing and Listening*. Lebanon, NH: University Press of New England.
- Spiegel, L. 1981. Manipulations of Musical Patterns. *Proceedings of the Symposium on Small Computers and the Arts*, Philadelphia, 19–22.
- Veinberg, A. and Noriega, F. I. 2018. Coding with a Piano: The First Phase of the CodeKlavier's Development. *Proceedings of International Computer Music Conference*, Daegu, Korea.
- Ward, A., Rohrerhuber, J., Olofsson, F., McLean, A., Griffiths, D., Collins, N., and Alexander, A. 2004. Live Algorithm Programming and a Temporary Organisation for its Promotion. In O. Goriunova and A. Shulgin (eds.) *Proceedings of the README Software Art Conference*.
- Waters, S. 2007. Performance Ecosystems: Ecological Approaches to Musical Interaction. Paper presented at Electroacoustic Music Studies Network EMS-07 Proceedings, Leicester, UK, 1–20. www.ems-network.org/spip.php?article278.
- Wiener, N. 1948. *Cybernetics or Control and Communication in the Animal and the Machine*. Cambridge, MA: Technology Press.
- Wiener, N. 1956. *The Human Use of Human Beings: Cybernetics and Society*, 2nd edn. New York: Doubleday.
- Wilson, S., Corey, K., Rohrerhuber, J. and de Campo, A. 2022. Utopia. <https://github.com/muellmusik/Utopia>.

VIDEOGRAPHY

- Haslam, G. 2020b. La Synthèse Humaine. *YouTube*. www.youtube.com/channel/UC6MtpOIZ7axVfJyA2oHytNg. (accessed 18 July 2023)
- Haslam, G. 2020c. The Bunker NY. *YouTube*. <https://youtu.be/pmEJXGOWw8E?t=520> (accessed 15 September 2022).
- Visi, F. 2021a. Global Hyperorgan: TCP/Indeterminate Place quartet + Piteå LTU Orgel Acusticum + Utopia Baroque Organ. *YouTube*. <https://youtu.be/0XV3IrB12tI> (accessed 15 September 2022).
- Visi, F. 2021b. TCP/Indeterminate Place Quartet – Hyperorgan Telematic Performance at NIME 2021 (excerpt). *YouTube*. <https://youtu.be/49UgqmFoNmE> (accessed 15 September 2022).