Composing with interactive music systems

In 1984 Joel Chadabe defined interactive composition as “… a method for using performable, real-time computer music systems in composing and performing music”. According to him, it involves two steps: Creating the system and composing/performing by interacting with it. In such cases, the creation and performance aspects are inherently entangled in both stages. Creating the system involves a feedback process of continuously testing and adjusting and interacting with the system in performance situations inevitably leads to ideas for further refinements. A system that is truly interactive requires at least two elements (usually a performer and some kind of software in computer music) that influence each others’ behavior. In other words, the software requires some input by the performer but also reacts to it in ways not entirely predictable, for example via a generative layer or a dynamic mapping scheme. Its sonic result then has some impact in the performers actions, who isn’t given absolute control. Reciprocal reaction is what distinguishes them from other kinds of systems involving digital musical instruments.

Composing, in this sense involves not only “the software that is written, the controllers that are used and the interaction that is defined” (Momeni, 1997, p. 2) but also the act of interacting with the system or playing the instrument. Thus, some of the traditional dividing lines between roles in music are blurred, such as composer/performer and instrument/score, The responsibility for the music composition and performance process is shared by the human performer and the software, while the later’s behavior functions both as the instrument and the score

Although we’ll describe some historical interactive systems that employ purely analog media, we’ll focus our discussion on systems for computer music. Therefore, whenever we talk about interactive systems we are really referring to interactive *computer music* systems.

An interactive computer music system involves one or many performers and a means of conveying information about their actions to a piece of software thas is ultimately responsible for the production of sound. This information if usually transduced via a physical device (which we’ll call the “controller”) into a set discrete data points that pass through a mapping layer, shaping the way they influence the resultant sound. The controller can be anything capable of producing data, examples are a couple of sensors attached to an acoustic instrument[[1]](#footnote-1) (the hyper-flute (Quintin, 2003) or overtone violin (Overholt, 2011)), mechanisms resembling an existing instrument (Piano MIDI controllers, the EWI), graphic interfaces on screens (the reacTable (Jordá, 2005)) or videogame controllers (Kinnect or Wiimotes). Unlike acoustic instruments, the sound producing mechanism is decoupled from the physical gesture, the latter only producing data in some format suitable to be mapped to parameters of some sound producing algorithm. The mapping scheme can involve unpredictable elements, such as parameters controlled by random number generators, patterns composing music algorithmically in real-time, or independent agents, such as machine learning models used to classify gestures or elements triggered via machine listening. It’s in this case when the system truly becomes interactive, as it involves real-time decisions being taken by at least two agents in response to each other.

One of the distinguishing features of interactive systems from electronic instruments is that the mapping layer involves some kind of generative approach. The system doesn’t simply allow a one-direction passive information flow, but takes the role of a musician in its own right, becoming a co-creator of the piece. There are various roles that this agent can take, including but not limited to those traditionally assigned to human musicians, such as performer, composer and conductor.

A case could be made to consider some acoustic instruments as interactive systems, as they tend to respond non-linearly or in a chaotic manner to energy input provided by a human. This can be attested by anyone that tried to learn a bowed string or wind instrument in a traditional western art music setting. The instrument appears to have a life of its own, creating sound in response more to the requirements of its physicality that to the urges of the novice performer. The instrument needs to be “tamed”, that is, the performer is required to be able to exert control over its sonic output. However, differences with acoustic musical instruments are many, the main one being that in interactive systems the performers rarely have has absolute control over the sonic result and is constantly in a kind of conversation with the system. The same input by the performer can generate radically different kinds of sonic behavior depending of the way the data is mapped.

In this chapter we’ll first explore the development of some of the first interactive music systems: Chadabe’s *CEMS* and Martirano *SalMAr Construction.*  Then, we’ll explore some of the elements that help us differentiate them from traditional instruments: the controller, mapping schemes and decision-making algorithms. The latter will lead us to a discussion about machine learning, a branch of artificial intelligence that helps computers identify patterns on input data, and thus opens new kinds of meaningful human-computer interaction.

**History: earliest interactive music systems**

Algorithmic thought and generative techniques in western art music have a long and fruitful history. One of the earliest known examples is Guido d’Arezzo’s combinatorial algorithm, used to set text to music by assigning two or three sets of notes in the 12-tone scale to a particular vowel, in a very similar way to how the syllables used in the solfège system were born. This is characteristic of abstract thought in music, where sounds are conceived not only as perceptual experiences but also as elements of a grammar. The modular nature of 12-tone equal temperament allowed for combinatorial practices to be commonplace in western music, with pitch classes and chords maintaining identity even with variations of register or voicing. Some examples include the 18-th century practice of musical dice game and the 20th century fascination with serialism. All kinds of algorithmic approaches have been explored, ranging from the unpredictable to the deterministic.

However, it wasn’t until the 1970’s that technology allowed for algorithms to run independently of human agency and respond to real-time changes. Early interactive music can be traced to the work of Joel Chadabe and Salvatore Martinaro.

At the State University of New York in 1969, Joel Chadabe installed the Coordinated Electronic Music Studio (*CEMS*) System, an automated synthesizer system designed by himself and built by Robert Moog. It consisted of three modular systems: Audio (oscillators, filters, amplifiers, noise generators, etc), Control (sequencers, envelope generators, mixers, etc) and Timing (a four-digit clock and 10 decoder/delays). Some of the modules were custom built and it became the largest concentration of Moog sequencers. The idea was to build a programable system that allowed control of independent but related parameters of sound synthesis by a single source. It probably was the first system that allowed for real-time algorithmic composition.

Soon after, he started sharing control of the sonic output by using joysticks as input device for his piece *Ideas of movement at Bolton landing* (1971). Any of the audio or control modules’ output could be shaped by voltage coming from the controllers. The result ended up being interactive: the system reacted to the joystick movements in ways not entirely predictable, while the performer reacted to the system’s output and tried to shape its behavior. He controller

Over the next decades he continued building and performing with interactive systems, starting to use digital media with his piece *solo* (1978). The system involved using antennas to sense proximity and scheduling sounds on a Synclavier using software. He could effectively conduct an improvisation of an orchestra of electronic sounds. This was in a way the conceptual opposite of Theremin’s *Thereminvox.* Instead of shaping individual sounds by controlling pitch and amplitude with left and right hand respectively, he shaped a whole piece by controlling overall tempo and timbre on real-time. Pitch and amplitude of every individual sound were left to be decided algorithmically by the software.

Simultaneously to the development of *CEMS*, composer Salvatore Martirano built an instrument called *Marvil Construction* with the help of engineer James Divilbiss. This proved to be a stepping stone in the development of a more ambitious interactive music system called *SalMar Construction*, which was finished in 1972 with the help of a group of engineers and graduate students from the University of Illinois, where Martinaro was a professor. The result was a 180-kg instrument, not including twenty-four loudspeakers and four subwoofers required for audio playback and spatialization. Its interface consisted of two sections. The lower was the main panel for live performance, consisting of an array of 291 touch-sensitive switches and lights to indicate their current state. The top consisted of a patching matrix to connect those digital control circuits to analog sound synthesis modules.

*SalMar Construction* could play 73 sound sources that were divided in four “orchestras”, basically interconnected sets of sounds patched in a way that they could share information coming from the performer via the state of the touch-sensitive switches. The way such information was modified by each orchestra could also be determined by such switches, so the logic of event scheduling by the instrument was almost completely unpredictable. The performer could loosely determine the overall texture of the piece and its general timbral distribution, switching anywhere from controlling all of the orchestras to changing the evolution of a single processes, but they always shared control of the resultant sounds with the instrument. The interaction devised for the instrument was analogous to conducting four different orchestras, each one improvising a concerto-style piece with its own soloist and ensemble.

The composer himself became a devoted and virtuoso performer of the *SalMar Construction*. However, he clearly wasn’t the only agent responsible for the piece, he could only make educated guesses as to what sound would result. “Control was an illusion. But I was in the loop. I was trading swaps with the logic. I enabled paths. Or better, I steered.” (Chadabe, 1997). Over the years he continued refining the *SalMar,* as well as composing and performing interactive music systems, such as the *YahaSALMaMAC Orchestra*, involving a Machintosh II computer running his SAL (Sound and Logic) software, a Yamaha DX7, multiple digital synthesis modules and Zeta MIDI violin, performed by Dorothy Martirano.

When working on the *SalMar,* Martirano wrote *Progress Report #1* (1971), a text describing the state the inner workings of the system. It ends with a short chapter consisting mostly of a series of questions concerning about the nature of real time topic, sometimes of a puzzling nature:

WHAT IS REAL TIME?

Those two four letter words have been used in this proposed report [many] times.

Does real time only exist when you think of it? Have you, who have skimmed through, thought of a better way to say it? Are you aware that the process that allows a real musical time to happen is a real musical? Where’s the trance? Can you sing and dance? Where’s the reflex? Is Wagner’s idea to put all the melodies together at the end of the overture less of an inspiration than the melodies themselves?

The best is A HEAD. (p. 84)

The emergence of a system seems paradoxical if we consider only its constituent parts. How does the organization of inert parts give rise to life? How is consciousness born from electro-chemical signals? When do discrete data points generate the illusion of continuity needed for computer music interaction? When does a thematic material become music? How are historically disjointed practices brought together to create a new tradition?

The meaning of the last (and rather criptically typed) sentence in the aforementioned chapter seems to be open for interpretation. Assuming the most literal, the best indeed was “ahead”, with pieces like Lewis’ *Voyager* (1987) and Rowe’s *Maritime* (1992) (to name just a couple of immediate successors) continuing with such developments. The next half century oversaw an exponential increase in the creation of interactive music and real-time composition/improvisation, aided by technological breakthroughs, the development of computer programming languages and sound synthesis software, and research on algorithms for machine listening, real-time digital signal processing and audio synthesis. Furthermore, the “entry fee” has been steadily decreasing. While the first experiments required institutional backing to see the light, powerful open-source software and cheap microcontrollers are commonplace now. Few are the prerequisites nowadays beyond a certain patience and frustration tolerance: while technology can be unwieldy at times it’s still within arm’s reach. A world of possibilities has thus been open, with a myriad of artists exploring anything from software for collaborative improvisation to interactive sound art installations.

**Controllers**

When using the word “controllers” in a musical context, I am referring to any kind of input device used for musical purposes. It’s the interface “mediating gesture and sound” (Roads, 1996), transforming information about physical actions from the performer to a signal suitable to be sent to a playback device, usually with an intermediate mapping layer that shapes it in some way. Such signals can take many shapes, from analog voltage control signals to discrete data points. The key difference is the transduction of physical gestures into electric signals. The differences between traditional acoustic instruments and controllers are manifold, and it could be argued that they are part of different categories: the first are integrated sound producing devices, while the latter form only the first step in the chain.

Acoustic instruments are easier to be perceived as a whole unit, each one forming an essence of sorts from where all kinds of sonic events can be brought forth into the world without them losing a fundamental identity. Even when we can dissect them to their constituent parts, these have roles that are interconnected, each one contributing in some measurable way to the overall sound. Particular configurations of material produce particular results, for example, it’s always possible to trace sounds produced by a piano to its original source. Even when considering extended techniques their timbre profiles tend to be limited to a vast but finite space of possibilities. The limit is not only determined by physical and mechanical constraints on the material or the arrangement of elements, but by the skill and anatomy of a performer. Moreover, the sound producing mechanism is the same as the instrument, with the energy provided by the performer’s physical gestures being transformed to sound.

In opposition, electronic musical instruments consist of at least two parts: a controller and a sound producing mechanism. They’re decoupled from each other and thus can be shared or exchanged by other instruments, as anyone with a cheap commercial digital synthesizer is able to experience by a simple change of patch. Furthermore, a one-to-one relation need not be maintained, multiple controllers could be shaping sounds on a single synthesis mechanism or the other way around. Instead of holistic systems they form entirely contingent systems, with no necessity shaping them and the specific configuration depending on the whims of the musician using them.

Of course, there is also a whole spectrum of possible designs between acoustic and electronic musical instruments, and multiple hybrid approaches exist. Everyone is familiar with electric instruments (think electric guitar), they are basically acoustic ones that require external amplification to be heard at loud volumes. Instead of sending information about a human gesture, soft sounds are converted to electrical signals that can be subjected to multiple kinds of processes, resulting in a wide array of possible transformations. Furthermore, there are extended instruments are acoustic instruments that are attached with sensors, and thus can send data to control sound synthesis or processing parameters in real time (some examples include the hyper-flute (Quintin, 2003) or overtone violin (Overholt, 2011)). Even though a case could be made to consider both as controllers, we’ll limit our definition to devices that generate data (usually via switches and voltage control) rather than audio waveforms.

Many controllers try to imitate shapes somewhat familiar to acoustic instruments and techniques like physical modelling synthesis and sampling can recreate their sonic counterpart, although none of this is a requisite and its barely a testament to their relatively new emergence and to a very human inclination for familiarity. It’s easier for an explicitly musical controller to be commercially viable if it has a smooth learning curve, therefore ensuring its adoption by performers and guaranteeing further refinements. Also, the functioning of such controllers is easier to grasp for the average concert attendee, ostensibly making the music more engaging.

On the other end there are novel or custom designs, or other kinds of controllers being adopted for musical use, such as videogame controllers like Kinect, a motion sensor device originally built as a peripheral for the Xbox 360 but that has evolved to become a commonplace device for many artists working on motion tracking. We can even find idiosyncratic designs in some of the early commercial examples, the Buchla Thunder being one of the most well-known. Nowadays, computers and tablets offer the possibilities of creating graphical user interfaces that allows us to employ them as musical controllers on their own right. The advantages are that novel designs tend to help generate new ways to engage with musical material, or sometimes a specific kind of controller is required for the way a composer envisions a piece. The reasons to choose between designs are numerous, and I’ve simplified the diversity of advantages of designs available. I have used traditionally inspired designs like the EWI[[2]](#footnote-2) to shape the overall evolution of a piece in my *Dasein* (2019) instead of playing individual sounds, and given enough time some original designs become commonplace, as exemplified by the Thereminvox pair of antennae. It ultimately depends on the piece or genre being played, as well as personal choices of the performers.

Some controllers come with predefined protocols used to easily communicate with computers. The most famous of these is MIDI (Musical Instrument Digital Interface). Created in the 1980’s by an effort of multiple instrument manufacturers to standardize a communication protocol for commercial digital synthesizers to communicate with each other. It encodes control data for musical performance, such as start and end times of individual notes, patch changes, pitch, and volume, but a flexible channel system allows routing any MIDI value to any parameter desired. The original protocol allows the passing of 7-bit control values, therefore its resolution is limited to 128 steps. MIDI 1.0 was so successful that it became the de facto protocol for communication between commercial controllers and music software, and it took around 40 years for it to be extended into MIDI 2.0.

Another protocol for music applications is OSC (OpenSound Control), developed at CNMAT, in Berkeley, California, and released in 1997. It has a higher resolution and flexibility than MIDI, allowing control data to be organized and routed in almost any desirable way. Each message can arbitrarily large and contain multiple data types: integers, 32-bit floating point numbers and strings[[3]](#footnote-3). It has been employed for client-server software architectures (like SuperCollider) and adopted as a control protocol by most DAWs, and for real-time interactive applications due to its low latency and ease of use. It is now employed for uses other than music, with fields such as robotics and visual art performance finding it useful.

On top of the obvious layer of interaction in real time performance of the system, continually developing and engaging with an interactive system over a long span of time requires a kind of interaction itself. It’s a process that involves two-way communication, allowing the possibility of feedback loops. Performing with them often suggest ideas for alterations, which themselves suggest new ways to perform, and so on. As described by Jorda (2005), “Music instruments are not only in charge of transmitting human expressiveness like passive channels. They are, with their feedback, responsible for provoking and instigating the performer through their own interfaces”. Even though alterations can be made at the mapping or synthesis algorithm layer, it’s influence nowhere more noticeable than in the controller design level.

Therefore, by trying to expand on existing models one is usually witness to the emergence of idiosyncratic approaches not only to controller design, but to music performance practices. Two paradigmatic examples are Michel Waisvisz *The Hands* and Laetitia Sonami’s *Lady’s Glove*, both used controllers build from scratch by the composers and developed at STEIM, a center for research on electronic performance located in Amsterdam, Netherlands. By employing different approaches to harness hand and arm movements as musical gestures they both managed to develop decades long performance practices that involved multiple iterations of the controller, numerous pieces, and a multitude of approaches to live performance.

The *Lady’s Glove* had 5 versions, constructed by Sonami from 1991 to 2003. It started as a humorous commentary on male-centered apparel in the design of controllers, placing some hall effect sensors in each finger and a magnet in the palm. It evolved to be an arm-long thin lycra glove equipped with all kinds of sensors: accelerometers, ultrasonic receivers, resistive strips, to name only a few. The analog signal is converted to MIDI, which is used to control anything from sonic material to motors and live video. Furthermore, she strived to control the music on multiple levels, from the individual sound to the structural elements of the piece. Being able to switch the focus on level and changing the degrees of freedom available to her, surrendering some control to the generative part of the system. This unlike Waisvisz’s *The Hands*,whose mapping scheme, as discussed below, allowed for more direct control of the sound

Even if the controller is built or approached with a set of ideas, the feedback process of design-perform engenders new sets. What started with an idea of feminist interaction design came through the years to incorporate issues of communication and embodiment. According to the composer herself, “… I realize that my imagination is pretty much molded by the system I use. I don’t think as much how will I adapt my ideas to the instrument, but I realize that the instrument has already influenced what I envision.” (Rogers, 2010). In such cases the evolution of the system is somewhat paradoxical, what can in retrospect be considered almost a teleological development into the current form is entirely contingent on the whims of the composer and their relationship with it. It requires agency but also kind of surrendering and close attention to the requirements and issues suggested by the system itself. The controller, being the most visible and tangible part but the least flexible, is usually the clearest path of communication where new ideas are suggested.

Of course, in live-electronics music performance there’s a third element involved in the communication: the audience. Sonami became concerned in creating a channel of communication that bypassed the opacity of many electronic music practices. It does this primarily by an evident concern with a directly embodied experience, with the attention paid to each gesture being the most obvious evidence. *Lady’s Glove* is a very clear example of a system that employs what Harrison et al. (2007) call the third paradigm of human-computer interaction. Focused not on human-computer coupling, and information processing and flow, but on “interaction as phenomenologically situated” (Harrison et al., 2007). This means that the system is focused on action as an activity that creates meaning depending on the context surrounding it.

Sonami’s performances with the *Lady’s Glove* can never be divorced from a multitude of elements that give meaning to it, including the interplay between the sonic output and the culturally dependent understanding of the meaning of certain hand gestures. Therefore, pieces like this have a fluid sense of identity that cannot be ascribed to a fundamental element or idea. They have the potential to truly become integrated into the moment-to-moment fabric of reality, meaning provided only (if at all) by a collective sense of interaction and temporal evolution. Embodiment itself then becomes the central focus, with the controller itself simply becoming a vehicle for the performance and potentially becoming translucent Situatedness takes a central role, and the controller turns into one thread in a truly tightly woven web. In Heidegger’s terminology, the controller turns from being present-at-hand (subject of enquiry, the focus of the performance itself) to being ready-to-hand (inconspicuous, a vehicle for the performance). This parallels the development of systems in general and *Lady’s Glove* in particular, with successive cycles of innovation becoming commonplace and providing the scaffolding for new variations to arise.

 I like the section on Laetetia, which you could contrast with a section on Michel.

I do think you could do more with Sal Martirano's report and think about contrasting

the various understandings of the significance of real time interaction different composers

project found in this work.

Mapping

The separation between input data coming from a controller and output sound allows and some times even requires an intermediate layer.

Constraints affordances

Murray browne

Algorithms

Composed instrument

My motivations

Teitelbaum article pg 3

Machine Learning

AI/animism

Heidegger’s Dasein

1. Also known as hyper, extended or augmented instruments. [↑](#footnote-ref-1)
2. Electronic Wind Instrument, a controller shaped like a woodwind. [↑](#footnote-ref-2)
3. Basically integer numbers, real numbers and text. [↑](#footnote-ref-3)