# 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] (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,

[1] Also known as hyper, extended or augmented instruments.

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 cryptically 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. In consequence, a world of possibilities has 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 so 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] 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

[2] Electronic Wind Instrument, a controller shaped like a woodwind.

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]. 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

[3] Basically integer numbers, real numbers and text.

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, was usually focused on 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 is charged with 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 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.

## Mapping

If we view interactive systems as a kind of information system, controllers represent the input. They are the way the performer uses their physicality to interact with the system, transmitting their energy and starting the information flow in the system. The output layer is usually some kind of sound generator, although it can take the form of other modalities, such as video. However, this model requires at least an intermediate layer between them, namely one that processes and transforms the incoming information in a way suitable to be used in some way by the output.  This is the crucial component takes the role of what we refer to when using the word *mapping*. A concept borrowed from mathematics; it describes the way elements of one set are assigned to elements of another.

The mapping layer determines the flow of data, and therefore what the user is allowed to control. Assigning values from one level to another is not an inconsequential task, as it determines the character of the resulting piece. Mapping can be considered stage in the creation of interactive systems where the developer of the system takes a role that is most like the traditional role ascribed to a composer. It’s the arena where artistic freedom can be most easily exercised, the only limitation being those inherent in the media chosen for the output. Any kind of mapping schema can be adopted and usually easily tested in real-time, thus allowing minute control over the characteristics of the system. Furthermore, the mapping layer usually works in a way analog to a composer, whose role usually demands determining parameters of control (pitch, duration, timbre, etc) that will be performed by the output element in the system (the performer(s)).

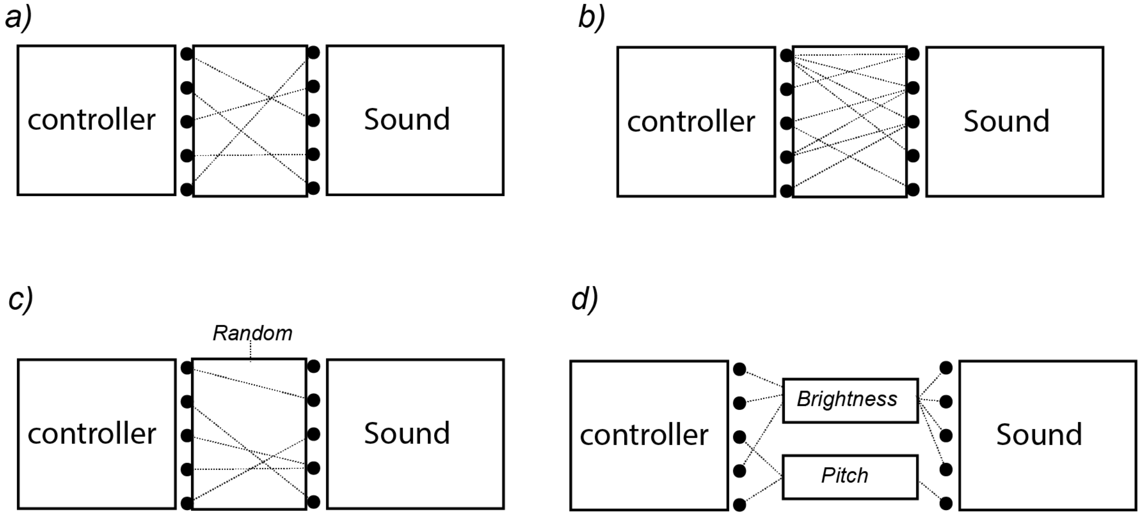
Mappings can be implemented in a multitude of ways, most of which fall in two categories: explicit and generative (Hunt & Wanderley, 2002). The first involves the composer of the system directly determining a set of rules that the mapping will follow. In contrast, the latter involves training a model that learns its own rules to associate controller inputs to sound synthesis parameters by providing paired examples of both. Machine learning approaches, such as artificial neural networks, are useful for such purpose and will be discussed in the next section of this chapter.

Explicit mappings are usually further categorized based on how each performance parameter is connected to each synthesis parameter, and therefore their correspondence. These are described by Hunt & Wanderley (2002) as one-to-one, one-to-many, many-to-one, and many-to-many. Figure 1 depicts one-to-one and many-to-many mappings (a and b). The middle rectangle represents the mapping layer, processing controller parameters and routing them to suitable synthesizer parameters.

One-to-one mappings are the most straightforward to implement, it’s as simple as mapping (in the mathematical sense) each input value to values in a range suitable for a particular synthesis parameters. For example, converting MIDI CC values (0-127 on a linear scale) from a controller to frequency (0-20Khz, on an exponential scale) values. This makes a lot of sense when dealing with parametric control of sinthesizers, but such simplicity paradoxically offers less control of the sound from a performers perspective, where dealing with multiple dimensions of movement at the same time is next to impossible. Only a few faders can be consciously moved at the same time, and controlling movement along 3 different axis for each body part in motion tracking controllers is next to impossible.

A study conducted by Paradis (described in Hunt, Wanderley & Paradis (2001)) explored user reported reactions when employing three different kinds of mapping, while keeping the rest of the system (a MIDI fader box and FM synthesis) unchanged. According to the researchers, when using one-to-one mappings “… many users noted that the simple division of parameters

*Fig 1. Examples of explicit mapping.*



*Note: a) one-to-one mapping. b) many-to-many mapping. c) dynamic mapping. d) multiple layers.*

was not very stimulating", while the most satisfactory was a many-to-many mapping scheme. Sound production is an inherently multidimensional endeavor, so when users are given multidimensional control by manipulating few parameters it makes synthesis more accessible to control. Concessions should be made by the system to accommodate the performer's embodiment, and rarely the other way around.

However, an even more interesting result of such study was that even if constant energy input wasn't a requisite for sound production, its introduction into the system made it feel "more natural.". Such extra input is analog to bowing or blowing on string and wind instruments respectively. By giving an extra measure of control that required constant movement and some time to learn, users were encouraged to explore with the system. Up to a point, we are interested by challenging activities. Exploring different mappings allows to find a middle way between interactive systems requiring a steep learning curve and being uninteresting, where user interaction becomes both playful and challenging.

Multiple and/or parallel mapping layers can be used which receive and control only subsets of inputs and output parameters (letter d in fig. 1). Called mapping chain by Arfib et al. (2003), this approach has the potential to make the mapping even more organic to human users by employing intermediate layers that map to psychoacoustical parameters of sound. For example, using "brightness" as a sound feature that can be determined by multiple controller parameters, such as lip and breath pressure in wind controllers (Hunt & Wanderley, 2002), and respectively determines multiple synthesis parameters, such as frequency and resonance in a low-pass filter or formant frequency and width in a formant oscillator. Letter c in figure 1 shows an example of a mapping chain with a single layer. Any kind of multilayer approach can be explored, for instance defining a space of features related to the controller as an intermediate layer before brightness in our example. Furthermore, such modularity makes it more flexible, allowing a single mapping scheme to be easily adapted for different controller/sound generator combinations. It's a way to allow more intuitive control and transparency in the performance by mapping using meaningful perceptual categories, instead of sound synthesis parameters, which are usually more related to the way the algorithm that produces them is implemented.

However, mappings don't have to be static. A multitude of approaches exist that can change the way the mapping work in what Murray-Browne (2012) calls *Dynamic* mapping. The simplest is to introduce randomness (letter c in fig. 1), either to shape the behavior of a mapping function or to control any combination of output parameters. This would involve the most basic requirement for a system to be interactive, responding to the performers' input data and shaping the output based on its own deterministic state. Implementations of randomness usually employ pseudorandom number generators, creating determinate sequences of numbers that simulate random numbers.

Dynamic mapping schemes can open new levels of affordances to the performer, or limit their feeling of being in control. On one end, switching between mappings could be another dimension of possibilities accessible to the performer, such as by changing the state of the controller. For example, pressing a button in a controller can allow to switch between "patches". In contrast, schemes could be designed that constantly interpolates or suddenly changes between different mappings, requiring the performer either to constantly adjust or give in to the inherently "experimental" character of the system.

Parameter mapping is not only musically useful for real-time digital instruments but is also a technique commonly used in data sonification. Sonification was born out of a need to create tools that exploit the auditory perceptual abilities for uses such as data analysis and information retrieval. The term “sonification” is understood as the derived techniques that deal more specifically with “…the data-dependent generation of sound” (Hermann, 2011). While it has been used for scientific purposes, it has also enriched and engendered new approaches to artistic sound creation and music composition. It is rarely used as an all-or-nothing technique by composers, but as one in a set of tools used to define parameters of the piece and relate the music to some extra-musical phenomena. Some examples of music built in part by sonification are Iannis Xenakis' *Metastaseis,* Alvin Lucier's *Music for Solo Performer*, and Charles Dodge's *Earth's Magnetic Field*.

*Earth's Magnetic Field* provides a great example of a composer using parameter mapping to determine the pitch content of the piece. Dodge used Bartel's “musical” diagram, a common method to display visually (in a way similar to the use of staves and notes) the average global geomagnetic activity, known as Kp-index[4]. He used the measurements of 1961 and mapped the 28 possible values of each “note” to a four-octave diatonic scale. This information was fed to a computer employing the Music IV software at Columbia-Princeton Electronic Music Center. This kind straightforward parameter-mapping sonification makes it possible to follow the melodic contour or sometimes individual notes throughout the whole piece, with Bartel's diagram providing the score for the piece. He maintained freedom to choose the actual character of the piece, as the data (while informing other aspects of the work) was used exclusively to provide the pitch parameters. He intuitively decided such defining aspects as the length and form. The timbre palette was freely chosen, loosely related to the phenomena that inspired the

[4] average global geomagnetic activity

piece, thinking of “radiant” characteristics and “the feeling of the human response… [to] the radiation from the sun that is essential to life” (Thieberger & Dodge, 1995).

This is an example of mapping as a “compositional process that engenders a structure of constraints” (Magnusson, 2010), namely that the relationship between pitch and solar radiation was fixed. The duality between affordances and constraints is an important mark in interactive music systems, as they determine the freedom of action given to the performer and therefore way they interact with it.

Affordances is a concept borrowed from the field of ecological psychology, defined as “a property of the environment that [allows] actions to appropriately equipped organisms” (Dourish, 2001). It’s the way an individual as situated in a particular context maps their potential actions to what is possible. However, it should not be understood as analogous to the controller/sound relationship that we have been discussing, with an intermediate mapping layer. Affordances are more integrated, related to the way we already exist as being-in-the-world and not to a duality between subject and object (with mapping provided by perception), the world reveals itself and it’s embedded in our embodied participation with it. In interactive systems, affordances are determined in the most obvious way by controllers. Depending on previous experience, a knob suggests the action of turning it, while a motion tracking device suggest a greater range of possibilities.

On the other hand, constraints determine a range of possible variations within which the performer can explore. If affordances affect the way we act with the world, constraints give shape to the extent of our actions. Inherent to all musical practice are a series of cultural and physical constraints, determining the limits allowed for a performer and defining the sonic space. While physical constraints are defined by the limits of human motion and the materiality of the instrument, and cultural constraints by what's expected in an environment defined by social relationships, interactive music systems allow for arbitrarily complex mappings to define what is *sonically* possible. Even if there are inherent limits in human perception, the potential to explore within them is to vast and ever more accessible via software. It's a self-imposed limitation, but it's what defines the character of the work.

If we conceive a piece of music as existing within a set of boundaries, and composition as carving a home for this entity in the multi-dimensional space of sound, then setting a series of constraints is where the compositional process is more explicit in the development of interactive systems. And mapping is what defines this set of constrains. As explained by Murray-Browne (2011) "In a time when musical programming languages have unleashed a bewildering amount of sonic potential, it is the constraints rather than the affordances of an instrument that characterize it" (p. 3). Mapping, as setting constraints, determines the state of the system and the kind of interaction the performer is most likely to engage in.

Interactive music systems emerge from affordances and constraints in a process of double articulation: affordances determine the elements to be used, while constraints select and organizes a subset of these in particular configurations, with the potential of giving rise to expression and/or meaning. We'll explore this type of dual emergence as evident in performances of *The Hands*, a custom-made MIDI-based digital musical instrument built and performed by Dutch composer Michel Waisvisz.

Waisvisz had a life-long interest in human touch as mediator of electronic sound, evidenced by his work with previous instruments, such as his *Crackle* series. Built in the late sixties and seventies, they consisted of devices built using analog circuitry in the form of oscillators, sensitive enough to finger pressure to elicit chaotic behavior with the slightest contact. With the performer sharing some measure of control with the unpredictable nature of the circuit, these stand together with *SalMar* and *CEMS* as one of the earliest interactive analog music systems.

He entered the realm of real-time computer music with *The Hands*, a device used to produce MIDI messages out of hand and finger movements which were used to control a Yamaha TX7 synthesizer. A first version of the instrument was built in 1984, with two iterations following in the next decades. They consisted of a controller strapped on each hand, with 12 buttons at fingers' distance, mercury switches detecting inclination of each hand, a potentiometer on the thumb, a set of ultrasonic transmitter/receiver to measure the distance between each hand, and a microphone usually employed to record and loop in real-time. Further technical details of its construction can be found in Torre et al. (2016).

Distance between hands was straightforwardly mapped to amplitude in most cases, a one-to-one mapping. MIDI note values, however, effectively formed its own mapping layer, with 12 buttons used to select pitch class, and hand inclination to determine the octave. This can be conceptualized as a many-to-one mapping using an intermediate layer. A "scratch" function (Waisvisz, 1985) could also be toggled on/off using the thumb potentiometer. Consisting of repeating a set of note-on messages (almost at audio rate) for each MIDI note being played when changing the distance between hands, allowing a kind of granular control of timbre using a gesture similar to bowing in string instruments. In such cases, this movement was non-linearly but directly mapped the overtonal content of the sound, using a deterministic dynamic mapping scheme.

In contrast to Sonami's *Lady's Glove*, *The Hands* were constructed to reflect Waisvisz interests in music performance as a display of physical effort and tension, a dimension lacking in most of live-electronics practices at that time. This resulted in the mapping described before, affording command in a way akin to how most traditional instruments are controlled by shaping sounds using direct energy input from the performer. Even the distribution of independent MIDI note messages to each hand, with the possibility of polyphony, suggests an expansion of control possibilities of keyboard instruments to include more dimensions related to timbre. Sonami's performances vary in the level of direct control she has over the sonic gestures, with that dimension being somewhat subservient to an interest in communication and the unfolding of meaning from hand gestures. Waisvisz's brought such control to the forefront of the musical discourse, shaping his whole aesthetics on a virtuoso display using physical effort as an expressive medium. Physical effort wasn't just an end on itself, but was a way for him to also show the "spiritual efforts made by the composer/performer" (Waisvisz, 1985). His performances usually went on anywhere between 30 minutes and an hour, resulting in him usually being drenched in sweat afterwards (Bellona, 2017). While both strived to generate a clear channel of communication with the public employing their extremities as main tools, Waisvisz's style focuses a more concrete and almost vicarious experience of somatic tension and release, while Sonami's is more concerned with abstract, semantic relations between gesture and sound.

Both the controller and mapping of *The Hands* set up a system of affordances. The first allowing for multiple dimensions of physical engagement, ranging from the intimate (buttons that require little movement) to the extensive (distance between hands), and the latter defining the almost instrumental-style control of sounds. This is the ecosystem where the piece was allowed to evolve.

However, he also set a series of constraints that helped shape such evolution. On the controller side, he consciously froze the development of the instrument for years at a time to focus on mastering its performance. The previously mentioned "scratch" function was an affordance limited by an inherent physical constraint, namely the extent of the performer's arms. The mapping, although somewhat dynamic, usually revolved around few parameters at a time, controlling individual sounds or the evolution of patterns. In contrast to the *Lady's* Glove, the computer rarely took generative roles and physical gesture was meant to be more tightly coupled to sound.

It's perhaps paradoxical that tighter constraints allow for more direct control, but when a limited space of possibilities exists the performer has more chances to map it (in a cartographical sense) and internalize its functioning.

This leads us to Chadabe's criticism of the concept of mapping itself to describe the coupling between the controller and sound producing mechanism in the context of interactive music systems (Chadabe, 2002). In short, he makes a distinction between two different approaches to control of a system, both related to different compositional approaches. It's like the explicit/generative mapping distinction (Hunt & Wanderley, 2002).

First is a more direct style of control, similar to Waisvisz close relation between physical movement and sound. It consists of a hierarchical (arborescent) structure, where control is mostly top-down. Even if the system shapes some of the resultant sound it is subservient to the emphasis on performance. It's here that we notice the prevalent tendency of digital instrument design to strive to be taken on equal footing as traditional acoustic instruments, trying to make its position unassailable by harnessing the prestige given to the "virtuoso" archetype in Western Art

Music[5].The importance of the human agent in the system is shown by giving the organic elements mastery over its silicon-based counterparts. It's analogous to an evolutionary adaptation of new forms of life trying to draw out of an already established ecological environment, looking for coexistence on equal grounds. It's in such cases when explicit mapping is useful, trying to keep tight constraints to allow more control for the performer.

In contrast, he talks about networks structures to explain what we'll call "generative mapping". This consists of a distributed network of control, where multiple agents share responsibility for the sound being generated, more akin to Sonami's approach. Instead of a hierarchical we get something similar to a flat structure. The human agent stops being the sole focus and becomes just a part of the system, its most visible and the one gifted with conscious decision-making, but a part, nonetheless. Arborescence turns to rhizome, with multiplicity of elements engaging in a conversation instead of a monologue. Emphasis is sometimes given to the performer, but more to their embodied experience as a node in an ever-mutating web than the center of it. This allows for non-virtuoso engagement with the system, and potentially stimulates the exploration of new modes of engaging with sound by setting an ever-changing set of constraints. The organism evolves by finding its own ecological niche but striving to eventually become a generalist.

This is where machine learning becomes useful. If we really intend for computers to take their own decisions in interactive systems, we can't limit to setting up a set of if-then rules. Teaching them to make inferences by providing hand-picked data is a way to steer their functioning without directly determining the constraints. Distributed structures can be achieved by employing literal "network" architectures for each node, such as artificial neural networks,

[5] For example, the theme of the International Computer Music Conference 2021 was "The Virtuoso Computer".

but many approaches to machine learning exist, even for deterministic approaches of control. In such way, arbitrarily complex and generative mappings can be created, and the idea of sharing control of the system with digital intelligent agents is brought one step closer to fruition.

## Machine Learning

In the years after its inception as a discipline in 1956 at a workshop in Dartmouth College, the field of artificial intelligence has suffered multiple false starts, owing to cycles of overpromising followed by the inevitable under-deliveries. Most of the early developments and breakthroughs focused on "symbolic" artificial intelligence, that is, trying to emulate intelligent behavior by manipulating high-level (symbolic) concepts, in a way that each step taken by the computer would be absolutely understandable to human agents. Expert systems basically consisted of a series of if-then computations that were supposed to mimic the way human decisions are made.

Even if some strides were made with this approach, the underwhelming results led to research on alternatives. They were further spurred by criticism of symbolic AI by philosophers such as Dreyfus (1986), focusing on the reductionist nature of symbolic approaches due to the assumptions made about the human mind working in such a mechanistic fashion. This positioned such work as rooted in a worldview biased by a Cartesian subject-object duality, where the ontological assumption was that elements of the world can be isolated from their context and manipulated independently. An intelligence was supposed to make inferences born of an ever-complex set of rules dictated from above, and not to learn from engagement with it's environment. The idea that objects can be abstracted from their spatiotemporal features is part of a deep philosophical underpinning in western culture, going back to platonic realism and its widespread adoption by the Roman Catholic Church. To escape from such constraints, we had to find a way to model consciousness as already embedded in the world and inseparable from its environment, allowing for a sort of subconscious thought. Nothing short from modeling intuition.

Thus, sub-symbolic approaches that employed statistics-based methods were developed in the early 21$^{st}$ century, giving rise to the field of machine learning. It consists in a series of algorithms that recursively improve their own functioning by learning from data.

It's impossible with our current means to model a completely embodied learning experience analog to human learning, but feeding an algorithm with data is the closest we can do at the moment. Even if the human agent is supposed to carefully select the data, choose the model, and tune a series of meta-parameters, the computer makes their own inferences at a level that makes sense for them, as proved by the "naivete" evident in adversarial attacks. For, example, where changes in some pixels in an image (undistinguishable to humans) can make a neural network classify it as something completely different. Even with these disadvantages, machine learning has been found useful for many daily life applications, anything from self-driving cars to music recommendation on digital platforms.

Machine learning approaches are usually classified as supervised or unsupervised learning, although there exist multiple techniques that don't fit neatly into either category. In supervised learning the computer is trained by giving examples of paired inputs and outputs, expecting it to learn to relate them in some way and produce its own outputs when given new inputs.

The most common uses of supervised learning are for classification or regression problems, the first giving discrete and the later continuous results. For example, a machine learning model can be to classify instruments in audio recordings. Training it would require feeding it with a dataset of features (power spectrum, MFCCs, zero-crossing rate, etc) extracted from recordings of a single instrument associated with the corresponding label (oboe, violin, etc). If properly trained, when presented with instrumental recordings or real-time feed from a microphone it will output the name of the instrument being played. Classification always distils the result to a single category, determined by an integer. In contrast, the results of regression can be any numerical value, including rational numbers. These are useful for cases such as creating a complex non-linear generative mapping between hand position and a continuous synthesis parameter, such as pitch. They are both basically complex functions, where the inner working is arbitrarily complex and learned by training using data.

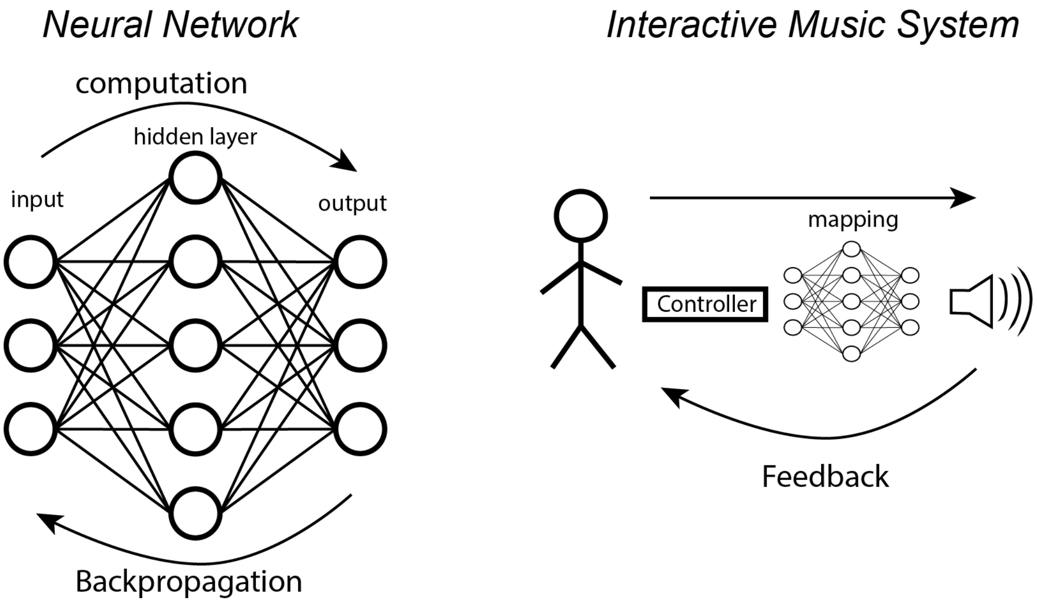
On the other hand, unsupervised learning requires no previous labeling or steering from a human expert, it learns underlying patterns in the data on its own and thus help give structure to seemingly disparate information. These structures can be represented in a way that is meaningful for the computer but not for a human observer and can be modified before reconstruction, common uses include text translation and denoising of images. This underlying representation of structures also makes them ideal for generative purposes, as it can create new instances with a similar data distribution to the original. It’s useful when the data shows no apparent correlation or when labeling is impossible due to the size or type of the database, such as when using unlabeled audio on a sample-by-sample basis.

In another chapter we’ll describe our experiments using unsupervised learning to generate material for electroacoustic music composition by interpolating between previously analyzed sounds.

While there are many machine learning models (decision trees, support vector machines, etc), artificial neural networks approaches have become particularly prevalent during the last decade, even giving shape to its own sub-field withing machine learning, that of “deep learning”. Its architecture is an attempt to loosely emulate the way neurons in the brain works as nodes in a network. Although not claiming to be an analogous representation of a biological process, it’s still an useful model based on developments on neuroscience. They consist of interconnected layers of artificial neurons, with an input layer receiving the data and output one calculating the result. Each artificial neuron is a unit of computation that receives numeric values from every neuron in the previous layer, usually weighting every connection differently, summing the result and passing it through a function to normalize its value before sending it to every neuron in the next layer.

Although at first this seems to be a one-way flow of information (with the feedforward approach previously described from input to output needed to generate the result of the calculation) this is far from the case, with result data flowing in the opposite direction for training purposes, in a process called “backpropagation”. It basically recalibrates the network to make more accurate result given the previous batch of computations. Its architecture is therefore

*Fig 2. Feedback in neural networks and interactive music systems.*



analogous to the feedback model we've used to describe interactive music systems, with a multiplicity of agents giving rise to it and the output influencing to the performance and the behavior of the system (fig 2). We can therefore consider generative mapping using neural networks for interactive systems as an example of self-similarity.

Admittedly, the way we have been dissecting interactive music systems into a set of constituents (controller/mapping/sound) can be seen as reductive, as is only a way to describe previously constructed systems. Even if they can be unthreaded for analysis purposes, music creation rarely engages consciously with them as independent of each other. A change in the mapping might be required by a modification of the controller, possibly changing entirely the affordance/constraints balance, and thus opening new dimensions of sound. Furthermore, when the relation between performance as a practice of embodied discovery and the shifting roles and relations with the system takes central focus, there's a need to explore alternative architectures to explore clearer relations between gesture/interaction/sound.

In Van Nort's article *Sound, Senses, Musical Meaning and Digital* performance, the author suggests machine learning is a way to engage with "phenomenological design work, allowing a more holistic approach to associating experienced/heard sounds to embodied and enacted gestures, collapsing listenting and acting in the design loop" (Van Nort, 2020). It would potentially allow us to bypass not just mapping as an explicitly defined layer, but even parametric control of sound synthesis itself by using methods such as Neural Synthesis (Engel et al. 2017), constructing audio directly on a sample-by-sample level.

Thus, the last decade has seen the evolution of machine learning tools for interactive art, such as the *Wekinator*, a piece of software developed by Fiebrink (2011) that makes multiple machine learning models and training accessible with a very simple graphical user interface. It receives data and outputs the results as OSC messages, thus flexible enough to be used with a plethora of controllers and audio synthesis software. With it, training itself becomes interactive, with real-time feedback in the system design process allowing for the creation of “complex relationships between performer actions and computer responses” (Fiebrink, 2020).