Topographies of Behaviours

Improvising with Spaces, Objects, and Algorithms

Slávo Krekovič

Master's Thesis

Institute of Sonology

The Royal Conservatoire of The Hague

May 2019

Acknowledgments

I would like to thank all the people who have supported me and helped me get this project done, but most of all: Dafna Naphtali, Guy van Belle and Jon Rose for their recommendations, Rebekah Wilson, Joel Ryan, Frank Baldé and Bjarni Gunnarsson for their generous support, guidance and inspiring conversations, Nicolas Collins and Richard Barrett for helpful thoughts and ideas. And last but not least, Kees Tazelaar, the whole Sonology staff and fellow students for a really unique creative environment.

Abstract

By investigating various concepts of abstract computational spaces, the notions of objects, ontology and agency related to the electronic music discourse, the research focuses on the development of an idiosyncratic interactive music performance system for expressive algorithmic improvisation. The system is implemented in the *SuperCollider* programming environment. Its main principle is based on topographies of generative behaviours in a low-dimensional parameter control space that facilitate dynamic morphing of sonic identities. This could be done in several ways: by navigating the control space and simultaneously influencing high-level parameters of the event generation via performer's gestural input, but also by making additional decisions about the operations with the sonic material. An important feature is a variable amount of autonomy of the system's response leading to the exploration of interaction possibilities in different scenarios, ranging from solo and "AI-assisted" performances to collective improvisations.

Contents

1 Introduction

1.1 Goals and Motivation

> *With the aid of electronic computers the composer becomes a sort of pilot: he presses the buttons, introduces coordinates, and supervises the controls of a cosmic vessel sailing in the space of sound, across sonic constellations and galaxies that he could formerly glimpse only as a distant dream.* (Xenakis 1992)

> *I've always been lazy, I guess. So I've always wanted to set things in motion that would produce far more than I had predicted.* (Brian Eno, as quoted in Shao et al. 2010)

This work is driven by the imagined "spaces of possibility". It is rooted in my 15-year-long practice as a musician mainly active in the areas of experimental electronic music and free improvisation as a genre. Although these two fields are quite close to each other, it could be argued that they have somewhat different historical roots, creative methods and partly also social milieus (Duch 2015). Anyways, my musical interests were always mostly focusing on the explorations in their overlapping area, emphasizing the "hear and now" (Wilson 2014) aspect of real-time music making more than composing fixed media pieces out of performance time. In my practice, music has been always a constant activity rather than the final artefact, similarly to how Adam Harper puts it:

> *[...] music is a socio-cultural ritual and not an art object. In a lot of "world music" music is something you DO, not something you listen to. You sing it, perform it, dance to it. This was still the case in Western classical music up until the beginning of the nineteenth century, when suddenly music was an artistic object you contemplated in silent reverence - but the ritual still applies psychologically even if physical participation was diminished. The illusion that music is an autonomous object is a very recent one particular to modern Western culture.* (Harper 2009)

The territory I am referring to could be essentially called "improvised experimental electronic music", whether it is about making music alone or with other people. The thinking about this approach has led me very naturally to some fundamental questions and issues related to the new instruments and performance systems facilitating innovative ways of musical expression. Such questions undoubtedly originate in the visions of artistic kind, but at the same time it would be unfair to deny the element of curiosity and fascination arising out of the current technological development. Although historically it happened many times that the ideas had preceded the then-available technological possibilities (the Futurists or Edgard Varèse would be good examples of this), part of the attraction of this research field lies in the fact that "*electronic music provides fascinating opportunities to build novel controllers and new musical instruments, or to expand the capabilities of traditional ones*" (Collins 2006). The design of interactive music systems and performance instruments has attracted me as a very vibrant and fascinating interdisciplinary subject area spanning different institutional and social contexts (in some of which I have been active in several ways): from universities and academically-grounded projects such as IRCAM or the NIME (New Interfaces for Musical Expression) conference, through unique artist-run platforms (STEIM with its unique position on the scene), to the DIY-practices and the whole independent cultural infrastructure comprising networks of experimental music and media art venues, labels, and festivals.

In my work as a performer I have been always interested in exploring the creative possibilities of different technologies including many hardware and software synthesizers and several programming environments. As an improviser with electronic instruments I have been often faced with the issues commonly mentioned in the NIME research, that could be essentially expressed as a polarity with a huge number of technologically-conditioned parameters used for sound genration on the one hand side, and on the other, a practical need of expressive and embodied interfaces ideally capable of quick and "holistically-driven" access to the desired sonic results in a real-time situation. But stepping one level up above these kinds of practical concerns, I have been also fascinated by the whole new worlds of possible musical interactions arising out of the fields of generative music, computational creativity and artificial intelligence research in general. Various machine learning methods and frameworks have become increasingly available even for artists and musicians without computer science degrees (such as myself), which created new spaces for exploration and incorporation of experimental technological approaches into their practices. It is evident that AI is already now changing the very nature of music making.

When asked about the motivations for the design of new instruments and performance systems, musicians and system designers commonly identify three main reasons (Emerson and Egermann 2018):

1. Wanting a more embodied experience when performing and producing electronic music.

2. Wanting to make the activity of performing electronic music more interesting to audiences.

3. Wanting to develop new sounds or timbres.

Moreover, as the authors of this study found out, during the research an additional motivation emerged:

> *The motivation to build responsive systems for improvisation that can surprise and challenge the performer certainly contrasts the motivations behind the development of most existing acoustic, electronic and digital instruments, for which the optimisation of performer control and the reduction of randomness or uncontrollability has been the focus. Using machine learning methods to introduce the "right amount" of randomness and develop systems that can be partners in musical improvisation marks a step in a very interesting direction for the future of music-making.* (Emerson and Egermann 2018)

It is exactly the fourth option that I find very appealing––but, because the improvisational nature of my work, the first and third ones are also relevant. The first agument for me relates to a much desired state of "flow" (Csikszentmihályi 2008) during the performance, which I consider very substantial although probably more difficult to achieve with digital performance systems than with traditional instruments. And although I would like to dispute the second, somewhat stereotypical NIME argument about audiences in the need of visual entertainment, embodiment has been always a vital part of my performances, where, ideally *"the instrument and performer may appear to dissolve into one entity."* (Paine 2009).

Speaking about NIME and the academic research in the digital instrument design subject area, I have to acknowledge that many of the conference papers were very beneficial to my research. But it is also a matter of fact that the findings of the researchers often remain only in the prototype phase and do not make it into the practice as tools available for musicians (Medeiros et al. 2014). Furthermore, no matter how interesting from a conceptual point of view, a review has shown that out of 78 multiagent system designs only 16 have made their software code available. This was yet another motivation for trying to develop my own performance system.

An attempt to design a music performance system is a challenging task opening doors to a large interdisciplinary research area informed by cybernetics, systems theory, embodied music cognition, human-computer interaction, philosophy. If this thesis was already not too long anyway, maybe this would be the right place to contemplate abut sonologists as present-day Leonardos, uniting many different skills and expertise from different backgrounds. While in the early days of electronic music there were often specialized engineers working together with the composers in the studios, we have to be sometimes not only composers-performers but also composers-programmers and composers-instrument builders:

> *Providing solutions that operate in realtime for concert performance is itself a further constraint. A composer must become a composer-programmer to have any chance of tackling this field. Such a modern composer abnegates the old 19th century image of composer-pianist, to become a blend of computer scientist, psychologist, acoustician and deferred musician.* (Collins 2006, 9)

The interdisciplinary nature of the field also means that the goals and motivations of the people involved in the designing of music performance systems could differ. Pearce et al. demonstrate this wide span in the Figure 1, refering to a broader field of algorithmic music composition:

Domain	*Activity*	*Motivation*
Composition	Algorithmic Composition	Expansion of compositional repertoire
Software Engineering	Design of compositional tools	Development of tools for composers
Musicology	Computational modelling of musical styles	Proposal and evaluation of theories of musical styles
Cognitive Science	Computational modelling of music cognition	Proposal and evaluation of cognitive theories of musical composition

Fig. 1. Motivations for developing computer programs which compose music (Pearce, Meredith, and Wiggins 2002, 128).

For myself it it evident that I the first line in the table is the most relevant, describing the case when *"computer programs are written by the composer as an idiosyncratic extension to her own compositional processes"* (Pearce, Meredith, and Wiggins 2002).

1.2 Research Aims

> *New instruments are not forced to remain at the sound and note level; with the added "intelligence" that computers can bring them, new digital instruments can also embrace algorithmic composition, they can deal with tempo, with multiple and otherwise conventionally unplayable concurrent musical lines, with form, they can respond to a performers in complex, not always entirely predictable ways (Chadabe, 1984).* (Jordà 2004)

One of the key issues that I have already mentioned in the previous passage (and it will keep recurring throughout the rest of this text) is the necessity of dealing with a huge number of technical parameters when designing and playing with digital musical performance systems. Practical concerns related to this have led me to the research of various methods of dimensionality reduction where I noticed that the concept of space was frequently popping up, taking on various meanings. Digital instrument design and electronic music theory, as well as engineering and computer science research often use this term when referring to spaces of different ontological nature: physical, perceptual, abstract and mathematical, cognitive, or computational. It was at first the Adam Harper's reinvention of the idea of an infinite music space with the myriads of variables, followed by conceptual search spaces as described by cognitive scientists and creativity researchers (Boden 2003, Gärdenfors 2000, Forth, Wiggins, and McLean 2010), and finally the notions of various feature and parameter spaces in the engineering discourse together with some practical applications that have provoked my interest in the spatial models as the basis for the development of an interactive music system. The metaphor of navigation in various kinds of spaces seems to reveal something fundamental about the ways humans think and act in the world, and as such it also very much relates to various kinds of representational methods for music analysis and building of creative tools. Based on these ideas I started to think about an abstract parameter space with different possible trajectories through it as a unifying design concept. As I will show later and more in detail, the concept is not at all new. However, from my experience with various "preset interpolators" and other music production tools utilizing the spatial principle I felt that there is more potential that needs to be discovered. Navigating timbral spaces could be just the beginning, but for an improvisatory performance the system would need to offer also some more generative and time-evolving features. After all, the spaces do not have to be at all static and free of (some kind of) life: what about populating them with objects possessing different kinds of agency? This is of course interesting from a theoretical and philosophical point of view; but how can it be brought down to the practical ground of real-time music making? How to use the computational spaces as productive playgrounds for musical interaction? What different types of spaces play a role in the design of music performance systems, and what strategies could be used to make efficient and fruitful connections between them?

Coming back to music itself, there were also some more general questions to be considered. What strategy could be applied to design a music performance system for an expressive and responsive control of the sound objects in an improvisatory situation? How to employ a palette of generative algorithms so that they can create a variety of sonic material controllable in real-time? How to achieve a variable degree of autonomy of the algorithmic procedures, possibly employing various kinds of software agents? Regarding the inclusion of the spatial aspect, these questions could be rephrased as follows: Could the spatial principle be a useful design approach for a system aimed at generation of fluid sound morphologies in real-time? What kind of interfaces and mapping strategies would be beneficial for achieving a variable (but manageable) degree of complexity in a live performance? How to create real-time controllable algorithms exploring the tensions and oppositions of static and evolving sound objects, regular and irregular pulsations, sparse and dense textures? In the following chapters I will try to provide some answers to these questions based on practical research findings.

1.2 Thesis Structure

The following text is structured into several chapters progressing from a more general introduction to the research field my views on the seminal topics to the description of the architecture, funcionality and possibilities of the performance system I have developed.

Chapter 1 includes this introduction, describing my goals, motivation and main research aims. In the Chapter 2 I give a general overview of my understanding of interactive music performance systems, investigating the issues of agency, specifities of computer improvisation and aesthetic questions related to the organisation of sonic material. Chapter 3 is devoted to the explorations of the different uses of the concept of space in the electronic music discourse and their possible relations. Chapter 4 features an overview of the performance strategies for the navigation in musical and computational spaces. It introduces various attempts of solving the dimensionality reduction problem in music performance systems and elaborates on the idea of parameter interpolation spaces. In the Chapter 5 I describe the principle of topographies of algorithmic behaviours and its application in the design of AMEN: The AMbiguity ENgine music performance system. Furthermore, I discuss the design considerations, explain the concept and architecture of the system, possible operations and technical implementation. Chapter 6 is final and it includes some reflections, evaluation of the system and conclusion.

2 Performance Music Systems

2.1 From Musical Instruments to Creative Systems

> *What characterises the new musical practices of the twenty-first century is a certain move from the linear format to the focus on the musical work as a system, an invention, assemblage or installation of sorts, whose materiality, spatiality, and situatedness separate it from the abstract notion of music in twentieth-century work, expressed as generalisable notes on the score, or "objectively" captured sounds designed for ubiquitous playback. (Magnusson 2019, 234)*

When summarising the tendencies in the evolution of tools used in the musical performance, Thor Magnusson gives an extensive and heterogenous overview of recent performance music systems that illustrate his observation of the changing nature of the musical work in the 21st Century (Magnusson 2019, 235). In his recent book *Sonic Writing* he traces the ways in which technologies have been changing the practices of music making and laying grounds for new compositional and performative approaches (Magnusson 2019). The evolution from linear thinking to the postdigital practices of building systems, networks or assemblages possessing generative abilities seems to be a crucial idea and forms the vital ground of my research. Magnusson distinguishes between the different modalities of what he calls sonic writing: the historically evolving manners in which humans use media and technologies in music making, from creating musical instruments (the co-called material inscriptions) and writing scores (symbolic inscriptions), through recording sound (signal inscriptions), up to creating interactive systems (digital writing). The accumulated historical knowledge gets iteratively inscribed into the new systems: the objects become epistemic tools, instruments of thinking, where "*the millennia-old distinction so clearly articulated by Plato between episteme (knowledge) and techne (skill) breaks down*" (Magnusson 2019, 9).

The latter development phase could be also attributed to a phenomenon sometimes called the ”realtime revolution” in art and music (Lopes, Hoelzl, and de Campo 2017, 343). As a consequence of an exponential increase of the computing power and affordability of the machines they became capable not only of generating complex sounds and images but also interacing with human or other agents in realtime. Following the previous revolutions (electronic, digital, and algorithmic), new programming languages, environments and hardware platforms such as laptops and microcontrolers with their niche social circles and institutions have accelerated this change towards novel musical practices. Yet Lopes et al. go further in describing an even more recent development they call the “autonomous agency revolution”, that *“has been quietly happening all along in the form of nontrivial machines, machines with idiosyncratic behavior, and other surprise generators”* (Ibid., 343). Although the origins of this approach could be traced back to the experimental analogue circuits of Louis and Bebe Barron, David Tudor and the like, the ideas of the systems designs inspired by the British cyberneticians have been commonly popping up in the recent discourse (Ibid.). Many of the new performance music systems expand on the 20th century concepts developed in the domain of algorithmic music composition (non-linear dynamical systems, generative grammars, simulations of natural processes, neural networks, etc.), translating them into the real-time and interactive domain.

2.1.1 Terminology

In relation to the “digital writing” paradigm, a whole new practice and research field called Musical Metacreation has recently emerged. Concerned with the automation of any or all aspects of musical creativity, Musical Metacreation *“uses the terminology of Generative Art (practice) and Computational Creativity (science) to cover autonomous systems of algorithmic music, generative music, machine musicianship and machine improvisation.”* (Tatar and Pasquier 2019, 7). Yet speaking of terminology in this domain, it is by no means standardised, and the new tools for musical expression have often different names: interactive music systems, extended instruments, composed instruments,[1] intelligent instruments, or meta-instruments (Fiebrink 2017), digital musical instruments (DMI), new interfaces for musical expression (NIME), and many more. The Robert Rowe’s classical definition of interactive computer music systems seems to be still valid, referring to systems *“whose*

[1] *“Schnell and Battier use the term composed instrument to signify the fact that digital systems can ‘carry as much the notion of an instrument as that of a score’ [...]. This highlights the fact – implicit in algorithmic composition – that computers can be used to predetermine specific structural aspects of a musical work as much as they can be used to realise as sound a musician’s action in performance.”* (Bown, Eldridge, and McCormack 2009, 190).

behavior changes in response to musical input. Such responsiveness allows these systems to participate in live performances, of both notated and improvised music." (Rowe 1992, 1). But over the years, interactive music systems could be found in several different contexts, comprising not only performances in concert settings, but also sound art or multimedia installations, net.art pieces, video games, audiovisual works, or dance performance pieces with interactive sound elements, with the boundaries being fuzzy. Although for the rest of this thesis I will focus on the use of creative systems specifically in a performance situation, it should not get forgotten that often the core of a system could be thought as universal, adapted to generate musical output in several of the mentioned contexts.

According to the *Journal of Creative Music Systems*, the computer systems of interest cover a broad area from systems capable of generating, performing and analysing music to systems capable of (online) improvisation, music-robotic systems, systems implementing societies of virtual musicians and others, including systems implementing computational aesthetics, emotional responses, novelty and originality.[2] In the following text I will narrow down the focus to the music performance systems, understood as a subset of creative music systems as a more general field. Although probably the most precise term for my research area would be "performance-oriented creative music systems", I will mostly use the short version "music performance systems". Sometimes the terms "creative music systems" and "music performance systems" are used interchangeably, but I am always accenting the live, real-time performance context.

2.1.2 Classification

An overview of the interactive music performance systems built in the last couple of decades would be a far too big endeavour and it was to a large extent done by other researchers (Tatar and Pasquier 2019). Nevertheless, it is evident that they could be placed on a continuous scale ranging from note-level control[3] (most synthesizers) through advanced sequencers (Laurie Spiegel's *MusicMouse* software, Frank Baldé's *The Lick Machine*, Jonathan Impett's *Meta-trumpet*) to full-fledged autonomous systems (George E. Lewis's *Voyager*, Robert Rowe's *Cypher*, *OMax* system born in IRCAM, François Pachet's *Continuator*, *Performance RNN*[4] neural network model developed by the

[2] https://www.jcms.org.uk.

[3] Including traditional musical instruments including – though the "note" notion can be disputed as maybe too much influenced by the classical lattice-based music paradigm –– consider cases with continuous control such as the Theremin.

[4] While there is a tendency related to the development of technology to perform more automatic or "intelligent" tasks in real time, with the level operation from symbolic to audio signal, even with new AI

Google Magenta team).[5] While some systems imitate existing music or at least try to keep their outputs in the boundaries of certain style (jazz, electronic dance music, etc.), others are meant to explore new territories and bring novelty into both the process and the results of music making.

There have been several attempts to classify performance-oriented music systems, but Rowe in his classic book *Interactive Music Systems* suggests a taxonomy which is still among the most relevant (Rowe 1992, 6–8):

1. *score-driven/performance-driven – precomposed versus spontaneous*

2. *transformative/generative/sequenced – suggesting the treatment of musical material and processing, the "composition techniques"*

3. *instrument/player paradigms – nature of the machine contribution, on a continuum from an extension of the human performance to an independent presence*

Regarding this view, my main interest lies in the performance-driven and generative systems as rooted in the previously mentioned postdigital practice of instrument building. However, speaking of the position on the instrument/player axis, the whole issue with the agency is somewhat problematic, so I will tackle it in a dedicated section.

I will now introduce three concepts which I find relevant to my approach to the development of a music performance system: the Creative Systems Framework, the method of bricolage programming and the theory of experimental systems.

2.1.3 Creative Systems Framework

The first is a formal tool that Geraint Wiggins proposed for modelling of the creative thinking, based on the theory of creativity developed by Margaret Boden. His Creative Systems Framework defines an exploratory creative system in a rather wide fashion as *"a collection of processes, natural or automatic, which are capable of achieving or simulating behaviour which in humans would be deemed creative"* (Forth, Mclean, and Wiggins 2008). The framework can be applied at various levels of abstraction, whether studying the details of inner algorithmic processes in a system, or observing the creative interaction of several agents (human and non-human) from a holistic perspective. Wiggins proposes a general formula describing such a system by defining the relationships between

approaches such as deep learning neural networks there are still systems that work with MIDI.

[5] See also Collins 2006.

several variables: the universe of possible concepts, subsets of acceptable concepts being evaluated according to the selected criteria, language to express them and rules to generate them. What he adds to the notion of a conceptual space as earlier introduced by Boden (2003) and Gärdenfors (2000) is the "traversal mechanism". The seminal idea here is condensed in an abstract spatial metaphor describing the generation of solutions by "traversing the conceptual space". However abstract this may sound, the spatial approach to thinking proves to be very useful for the understanding of cognitive mechanisms in relation to music and building analytical tools. The spatial metaphors will be frequently popping up in the following text for a good reason and as I will try to demonstrate, they can be applied as a central mechanism for the construction of creative systems. As Wessel and Wright mention, *"one of our central metaphors for musical control is that of driving or flying about in a space of musical processes. Gestures move through time as do the musical processes".* And while this framework is supposed to be a general formal tool for the modelling of cognitive processes, several authors have successfully applied it also in the domain of music (Tubb 2010).

2.1.4 Experimental Systems

The last idea important for my development of a music performance system, and probably widely relevant in general, is Hans-Jörg Rheinberger's theory of experimental systems. Although originally developed in the scientific context of biology, its author himself was the first to suggest the application of the principles in the artistic domain, by suggesting that writing in itself is already an experimental system (Schwab et al. 2013). Summing up Rheinberger's own different definitions,[6] experimental systems can be understood as the smallest functional units of research, "loosely coherent" assemblages of conceptual, technical, social and other conditions for giving *"answers to questions that we are not yet able to formulate clearly. In a typical case, an experimental system is … a 'machine for making the future.'" (Rheinberger 2015).* An important feature an experimental system is its ability to lead to discoveries by yielding surprising results, because "*one never knows exactly where an experimental system will lead. As soon as one knows exactly what it produces it is no longer a research system."* (Ibid.) Such setups involve the use of technical objects to set up conditions for the production of knowledge. In the context of creative disciplines this would mean that experimental systems can be used to produce novel aesthetic approaches as a special kind of artistic knowledge,

[6] "'*Experimental systems are arrangements that allow us to create cognitive, spatiotemporal singularities'*, '*experimental systems are machines for reducing complexity'*, or, also: *Experimental systems are hybrid arrangements: in a permanently fluctuating and varying pattern, they mix up elements which many historians and philosophers of science, and sometimes even scientists (at least in their semi-popular essays) wished to have properly separated."* (Rheinberger 1994, 359).

expanding the established field of practice in ways that would be not possible otherwise. In the sphere of performance music systems development, this would be partly the kind of methodology described by composer, artist and programmer Hanns Holger Rutz as "experimentation with the algorithms" (Rutz 2016b): "traversing the space of solutions" by the means of bricolage programming.

Paulo De Assis adds an interesting point regarding the use of technical objects in the experimental practice, which are bridging the past and the future:

> *One might say that scores, instruments, or tuning systems, for instance, may be seen as technical objects that are brought into particular constellations (such as "the concert" or a CD recording), to produce art. The same entities may, however, operate as epistemic things, whose qualities can be divided into two main groups: those already known and those still to be known (discovered). Musical works participate, therefore, in two different worlds: one related to their past (what constitutes them as recognisable objects), another related to their future (what they might become).* (Schwab et al. 2013)

I can see a difference between the experimental system and the music performance system in that the development of the latter is taking place in the framework of the former, which is usually happening outside of the "performance time". But, in many cases it can also overlap with an actual performance that could be therefore also regarded as an experimental system in which the audience plays an important role.

When Rheinberger states that *"for a long time, experimentation did not occupy a prominent role, neither in philosophy, nor in history of science"*, it could be added that the same was true for musicology that was for a long time focusing mostly on the structure of musical works and not so much on the creative process. He then goes further by describing how the social adoption of the experimental systems lead to the formation of "experimental cultures", which could be also said about the vital movement we could observe in the avantgarde music of the 20th and 21st Centuries.

2.1.5 Bricolage Programming

When trying to answer the question "What is the relationship between an artist, their creative process, their program, and their artistic works?", Alex McLean and Geraint Wiggins (McLean and

Wiggins 2010) suggested the term “bricolage programming”. By doing so, they actually “refurbished“ the conclusions of earlier observations by Sherry Turkle and Seymour Papert who had been describing programming style of two college students as a bricolage (a term originally introduced by Claude Levy-Strauss in 1962).

> *The bricoleur resembles the painter who stands back between brushstrokes, looks at the canvas, and only after this contemplation, decides what to do next. Bricoleurs use a mastery of associations and interactions. For planners, mistakes are missteps; bricoleurs use a navigation of midcourse corrections. For planners, a program is an instrument for premeditated control; bricoleurs have goals but set out to realize them in the spirit of a collaborative venture with the machine. For planners, getting a program to work is like “saying one’s piece”; for bricoleurs, it is more like a conversation than a monologue.* (Turkle and Papert 1990, 136)

The dialogical concept seems to be very relevant also in the world of computation, art and the design of creative musical systems. The feedback loop in the creative process is based on perceptual evaluation of tested algorithms.

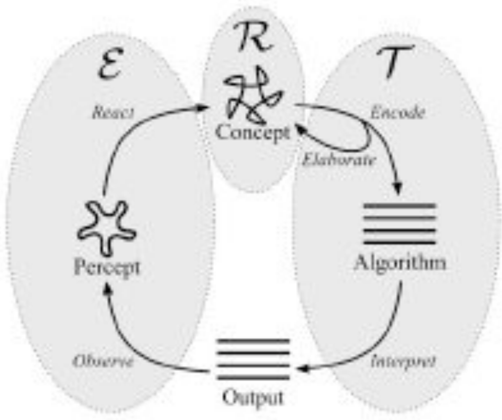

Fig. 2. The creative feedback loop (McLean and Wiggins 2010).

The depiction in the Figure 2 is an application of the Creative Systems Framework methodology: both the perceptual and conceptual levels have spatial attributes, and creativity is characterised as a search in a space of concepts:

The programmer's concept R motivates a development of the strategy T to be encoded in a program, but the programmer does not necessarily have the cognitive ability to fully evaluate the program. That task is taken on by the interpreter running on a computer system, meaning that T encompasses both encoding by the human and interpretation by the computer. The traversal strategy T is structured by the techniques and conventions employed to convert concepts into operational algorithms. These may include design patterns, a standardised set of ways of building that have become established around imperative programming languages. Each design pattern identifies a kind of problem, and describes a kind of structure as a kind of solution. (Forth, Wiggins, and McLean 2010)

From my own experience, the authors' point seems highly appropriate not only when referring to the process of development of algorithms, but also when describing the workflow of the testing and connecting of different hardware devices, platforms, programming environments, software's APIs (Application Programming Interfaces), or communication protocols. Moreover, the bricolage-like nature of development has been especially made possible with the concept of interactive programming, facilitated by interpreted languages like SuperCollider or environments for visual coding such as Max and Pure Data.

2.1.6 Idiosyncratic Performance Systems

The artistic and creative activity in music in general––and even more so in connection with the technology––is a highly experimental field where people have the ambition of finding unique solutions and because the arrival to the solutions in the systems-based music of the 21st Century is now seen as the compositional practice itself. Therefore, there is no, and cannot be, a standard way of approaching the design of creative systems for musical performance. The boundaries between composer, performer and system designer are disappearing which creates a new, highly unique field of creative activity.

Coming back to Magnusson's list of musicians, it is astonishing how many radically different approaches could be found just in the very recent period: from the use of no-input mixing boards, circuit bending approaches and creative adaptations of existing instruments through networked and robotic performances, simulations of organic behaviours, to interactive real-world data sonification examples, all these systems create a set of very individual takes on the systems building. Is it the elegance of the design of a system, the satisfying feel of interaction with it, or, in the end, the quality and range of possible music it can make that matters? Instead of exercising computational formalism of creating models demonstrating certain principles, I would like to embrace the notion of an idiosyncratic system, with all its possibly bricolage-like and hybrid nature, because *"unorthodox cross-connections can create forms of musical expressivity that can only emerge within such idiosyncratic systems. Here, investigation means experimentation and navigating towards the unknown."* (Lopes, Hoelzl, and de Campo 2017, 343).

In addition to that, as varied and complex in their structure they may be, it should be noted that these systems are mostly not fixed and definite solutions. Instead, just as every performer and composer evolves in their skills and aesthetic preferences, their systems evolve with them. Sometimes they may be created as unique setups for one occasion, other timer they develop stepwise with every iteration of a performance. All this underlines their ephemeral nature: the performance systems, "composed instruments" or any other experimental setups, are thus momentary configurations of objects, temporary assemblages of ever changing constellations contributing to the "post-instrumental" musical practices (Ibid.).

2.2 Object-Oriented Ontology

> *After the different waves of cybernetics, after information and control theory, semiotics and linguistics, cognitive science and artificial intelligence, we are reaching a point that, depending on the school of thought, could be characterised either by an intensified constructivism, or by a renewed realism, both of which de-emphasise the human subject and the categorial split between humans and machines.* (Rutz 2016b, 73)

With the aim of going a bit beyond the technical language of the human-computer interaction domain I would like to foreshadow some of the background remarks, illuminating my view of the design of interactive systems.

While there has been a strong tradition in the style of Leibnitz to make a sharp distinction between the “substancies” (subjects with a soul, or, as we might also call it, consciousness) and “aggregates”, inorganic objects such as machines, there are increasingly more thinkers that advocate for a less anthropocentric view of technology as being more appropriate in today’s world. When admiring Bruno Latour’s critique of modernism, the philosopher Graham Harman (Harman 2018) uses a nice comparison: the goal of modernism was to quarantine humans from nature to keep culture and nature as separate domains, pure as possible. But, even Latour’s solution to deconstruct such concepts and name them as ”hybrid” has to be abandoned for its actual acknowledgment of this false dichotomy. When referring to complex things, under which also creative music systems could be categorised, Harman introduces the term *compounds* that seems to address a mixture of their qualities that can be both natural and part of the wider cultural context. An example would be a machine such as, in the musical context, a synthesizer: it has undeniable physical properties since is made of molecules and floating electrons, but it is also as objects with certain cultural significance and embodying some previous collective knowledge according to Magnusson’s notion of epistemic tools.

Harman's Object-Oriented Ontology,[7] loosely inspired by Bruno Latour's Actor-Network Theory framework (ANT) of interconnected actors of arbitrary origin (human or non-human) suggests an intriguing point of departure that can be, in our case, used as a philosophical framework for describing the human-computer interaction. In the OOO terminology, everything can be regarded as an object, which is a synonym for entity or thing. Objects exist on different scales: they can be humans, animals, robots, fictional characters, corporations, galaxies, programming languages, oscillators and filters, as well as abstract geometrical structures such as curves or spaces. So, how can an object be defined? A comprehensive description of an object of an arbitrary complexity, in respect to its emergent properties, is that it is "more than its pieces and less than its effects", in other words, it's not reducible neither "upwards" nor "downwards". But perhaps the most important characteristic of the OOO theory is that *"all objects must be given equal attention, whether they be human, non-human, natural, cultural, real or fictional."*[8] I will come back to this in the next passage when discussing the problem of agency. The rather static appearance of the OOO view might be misleading, since even events, relations and processes of various scales can be considered objects, be it a hurricane, relation between an input gesture and a sonic output of a musical system, or a crash of the computer's operating system.

Harman makes an important distinction between the "real" (as it is) and "sensual" (as we perceive it) object with its real and sensual qualities, which alltogether form a quadruple of tensions. This is a view that can be relevant in the understanding of aesthetic principles but also how in explanations of how computational systems work. For instance, an algorithm is a real object with real qualities. But we (or any other object) can only experience it indirectly, as a sensual object with some sensual qualities perceptible to us. The algorithm––or any other object––can be considered a "black box"[9] we can only interact with through its "inputs" and "ouputs" (if it has any). But a textual representation in the form of a programming code is also an indirect one. Various other objecets taking the form of visual representations in the form of Graphical User Interfaces (GUIs) or debugging messages have to be used to track at least some information the its qualities. As I will try to demonstrate, OOO can be a relevant tool for the thinking about music performance systems in several ways:

[7] Inspiring views similar to Harman's are also shared by Manuel DeLanda, Ian Bogost and Timothy Morton, among others.
[8] The object-oriented turn makes a departure from the dominant part of the tradition of Western philosophical thinking which had the tendency to eliminate objects by either undermining (reducing phenomena to the smallest components such as atoms) or overmining them (what we perceive is not real, it is only processes, events, surface-effects that are real). See Harman 2018.
[9] There is more about the black box nature of algorithms in the text of Bjarni Gunnarsson. (Gunnarsson 2019).

– Studying and modelling interactions between objects of various nature and with different agency on several levels of the system (human and software agents, physical objects, virtual objects, etc.).

– Comparing the evolution of sound objects as perceptual entities with the behaviour of algorithmic objects that have correlates in the “parameter space”.

– Researching objects as interacting software components. In the object-oriented programming languages the parallel is clearly perceivable, but this view can be essencially applied to any algorithms.

For instance, the music performance system I have developed was mainly written in SuperCollider which is also an object-oriented language:

> *All entities in the language are objects. An object is something that has data, representing the object's state, and a set of operations that can be performed on the object. All objects are instances of some class which describes the structure of the object and its operations. Objects in SuperCollider include numbers, character strings, object collections, unit generators, wave samples, points, rectangles, graphical windows, graphical buttons, sliders and much more.*[10]

A further investigation of sound generation algorithms as objects can be found e.g. in Bjarni Gunnarsson’s work (Gunnarsson 2012).

It has to be mentioned as well, that in the theory of sound-based music, the sonic or sound object is a well known category. It is traceable back to the extensive theory of sound objects conceived by the electronic music pioneer Pierre Schaeffer (Schaeffer 2017), whose terminology has been widely accepted.[11]

---

[10] http://doc.sccode.org/Guides/Intro-to-Objects.html. Accessed 10 May 2019.

[11] More about the ontologies of sonic objects in the *Handbook of Systematic Musicology* (Bader 2018, 761).

2.3 Agency

> *Algorithms now bear a crucial relationship to material reality, they can have unintended consequences, they can crash machines, etc. Algorithms have become performing entities. This has profound aesthetic consequences.* (Rutz 2016a, 31)

In the reseach about the principles of performance music systems, a topic of agency tends to be recurring. In the human-computer interaction it is a common task to investigate not only *what is happening* but also *who (or what)* is doing it and how they are taking part in the music making process: acting, reacting, moving through space and performing gestures, generating sound, listening, making decisions and changing directions. The issue is, of course, even more prominent in the music performance systems that specifically aim to generate nontrivial and autonomous behaviours. However useful, it should be said that the concept of agency in computer music performance––somewhat related to the concepts of causality, but also indeterminacy and responsibility––is far from unproblematic.

First and foremost, in the interactive systems (but this is true for all software) agency appears to be present on different levels and in various scales. For the start, let us not take into account human actors, even if they should be considered as parts of the system. It is, for instance, often stated that the whole interactive system is an agent (Collins 2006, 26). But since objects can be composed of other objects and each of them can be performing their own agency, the same could be said for some of the system's components, such as algorithms: just imagine a procedure generating a probabilistic control process that has influence on another process, reacting to the input from an external object, etc. It is quite possible that in such systems even a small piece of code could substantially influence the whole musical processes and thus have a degree of what is called "performative agency" (Bown, Eldridge, and McCormack 2009, 194, Sanfilippo 2017). Even if we would agree to admit a deterministic and predictable nature of the software components, which is questionable, their interactions may create a spontaneous and surprising behaviour (Evens 2006). This is what could be called a distributed agency (Hayles 2016) on a low level, although the principle could be also scaled to include constellations of larger objects.

Stepping one level up, the autonomous agents are usually defined as more high-level "*autonomous pieces of software which contain perception and action abilities.*" (Tatar and Pasquier 2019). What exactly are these abilities? Nicolas Collins debates the relevance of the criteria set for the distinction of autonomous agents as set by in contrast to mere software. According to a definition quoted from Franklin and Graesser, an autonomous agent is a "*system situated within and a part of an environment that senses that environment and acts on it, over time, in pursuit of its own agenda and so as to effect what it senses in the future*" (Collins 2006). One of the newer definitions states, similarly, that

> *[...] an agent is an autonomous system that initiates actions to respond to its environment in timely fashion (Wooldridge 2009). Similarly, musical agents explore the notions of autonomy, reactivity, proactivity, adaptability, coordination and emergence. (Tatar and Pasquier 2019)*

The latter description is even more narrow in that it makes a tight connection between the notion of an agent in general and its autonomy. While there should no issue with framing many of the creative systems as being part of the environment and acting on it, following "its own agenda" seems to be a disputable task. Is "waiting for an input from a human performer" the own agenda of a listening agent? The proactivity condition is even more tricky. Are there ways any action can emerge in the running code without being explicitly told so in advance? Most probably there are, especially in the case of the indeterminate nature of more complex algorithms. But in terms of tracing the actual amount of agency and the level of decision making, Collins made an interesting point in trying to see interactive music systems as being parasitic on human musicians, *"who must willingly close the feedback loop to let the system's actions change future situations. If we accept the musician's willingness to enter such a contract, demanded by conventional concert etiquette, then the interactive music systems may fall under the kinder interpretations of 'autonomous agents'" (Collins 2006, 9).* The extent to which pieces of software code are autonomous in their behaviour from the goals of their programmers remains a philosophical issue related to the concepts of unpredictability, randomness and, last but not least, the emergence of musical meaning. However, this is not to say that algorithms cannot be considered creative, as can be seen, for instance, in the recent deep learning neural network models.

2.3.1 Artificial Agents

Let us focus on some more differences between the possible artificial agents as defined by the research of interactive systems**.** Wiggins, for instance, sees them as entities with a different degree of autonomy:

> ***Stateless Agents*** *have no memory and merely respond reflexively. Example: a thermostatic heater.*
>
> ***Agents with Memory*** *have internal state, and can therefore reason from information about past states of the world; they are still essentially reflexive, but better informed. Example: a TiVo unit that automatically records a favourite series, once its user has recorded two episodes.* ***Agents with Goals*** *have aims beyond merely responding to individual stimuli; their responses to individual stimuli are conditioned by their longer-term considerations. Example: a driver-less car that has no map, but works by following the road and using a compass to determine which direction to turn at each corner.*
>
> ***Utility-based Agents*** *use utility theory to act rationally, achieving their goals in ways that might be said to be more efficient according to a given measure. Example: a driver-less car that gets its rider home by the quickest and most fuel-efficient route by using maps and live traffic information.*
>
> *Missing from this list is the idea of* ***predictive agents****. While it is certainly the case that a Utility-based Agent, and possibly an Agent with Goals, will plan ahead to determine a route to its goal, this is a different kind of prediction from that intended here.* (Wiggins 2018a, 21)

Collins and Kusch suggest to distinguish another four hierarchically ordered machine types with various levels of behaviours:

> *1.* ***Behavers*** *instantiate exact repetitions*
>
> *2.* ***Disjunctive Behavers*** *can act as different behavers based on an analysis of the thing to be operated on*
>
> *3.* ***Feedback Behavers*** *respond within a range rather than a discrete set of behaviours and are able to cope with novel stimuli*
>
> *4.* ***Learning Behavers*** *can learn from users and the environment to modify their behaviour (Collins 2006)*

As Collins notes, *“the interactive music systems are at most feedback behavers, in that they can respond to stimuli within a predetermined range, and their behaviour is not entirely predictable in advance*“ (Collins 2006).

To add to the overview agent-based interactive music systems, in their recent and very extensive review Tatar and Pasquier mention three types of artificial agents: 1) pure computational software agents, 2) agents defined by computer generated image (CGI) ”living” in a virtual environment, and 2) robotic agents with a physical body. They have reviewed 78 purely computational agent systems and classified them based on their “level of intelligence” again into three categories: cognitive, reactive and hybrid musical agents as illustrated in Figure 3 (Tatar and Pasquier 2019). When assessing the degrees of the agents’ autonomy, the results are––unsurprisingly––positioned in a continuum between reactive and completely autonomous. Furthermore, the authors elaborate in depth on the categorisation of several other attributes of the systems such as the internal architecture, number of roles they can take, type of environment, corpus as a kind of working material, musical tasks it is capable of doing, or way of external communication.

Fig. 3. Taxonomy of multiagent systems (Tatar and Pasquier 2019).

Interestingly, as can be seen from further evaluation of the systems, quite often the nature of a system in particular category would be described as hybrid. This can be attributed to the fact that musical systems for a real-world practical use are rather complex, utilising several methods of interaction design, and more often than not, they are not merely simple demonstrations of one design principle or an idea.

Since this is probably the most complete typology up to date, I will get back to it when diving into the details of my performance system.

2.3.2 Behavioural Objects

Without trying too much to accent a posthumanist perspective, in a world increasingly dominated by computational ecosystems, with the ubiquitous automated data surveillance, smart banking, location-aware services or targeted election campaigns it seems quite necessary to be able to overcome the anthropocentric view of a human-computer interaction. As mentioned before, when we look at the history of live algorithmic music, anthropocentric descriptions of various interaction concepts seem to span across a wide spectrum, ranging from the notion of an ”instrument” to an autonomous ”player”, as in the case of Rowe’s typology. Within the so-called “acoustic paradigm” referring to the Western classical music idioms, “*clear roles are defined for both the people (performers, composers and luthiers) and objects (scores and instruments) implicated in these activities”* (Bown, Eldridge, and McCormack 2009, 188). However, more critical approaches trying to philosophically reframe this creative field advocate for overcoming the anthropomorphic perspective, and offering a more flat ontological understanding of the interaction. In concordance with the intentions of OOO theory, by attributing a more active role to the computational medium, Oliver Bown and his colleagues offer an interesting step away from the acoustic paradigm. Instead of software instruments, artificial improvisers, machine listeners, and score-followers they suggest to use the term “behavioural objects”:

> *This concept is aimed at emphasising the active nature of software (its behaviour) at the same time as its role as a tangible unit of social exchange, and as a creative tool (i.e. as an object). Behavioural objects can potentially exhibit complex behaviours like machines and organic structures, but can also be exchanged between people as rapidly and effortlessly as ideas.* (Bown, Eldridge, and McCormack 2009, 188)

The authors claim that it is the development of the software that is causing the the traditional roles and distinctions to change. Behavioural objects can be pieces of computer code with the ability to react to information from both within the software and from the outside world. Translated to the field of creative systems, this view is acknowledging that the responsibility for the musical result gets divided among he performer and the software, both performing variety of actions. Coming back to the topic of this section, behavioural objects are capable of two kinds of agency: the performative and the memetic. While I have already discussed the performative agency, the memetic agency is an appealing concept describing the the ability of a software system to influence––in OOO terminology––another objects or hyperobjects, such as musical styles, across a larger time period and thus to become a “driver of cultural change” (Bown, Eldridge, and McCormack 2009, 194). In this way, the memetic agency can also establish conditions for performative agency: let us imagine the software that facilitated the development of live coding practice as a new genre.

2.3.3 Other Objects

Whereas the anthropocentric view manifested in the “acoustic paradigm” with its composers, compositions and instruments could be overcome by the introduction of behavioural objects, much of the discourse about the agency in the performance music systems again totally omits agents other that human and machines. And although it is understandable that the human-computer interaction (HCI) research focuses predominantly on what could be called computational agents, I believe that the creative artistic and musical practice brings many examples of agency where a broader and less technologically-biased view could be useful. To bring in an example from my own practice, during the development of my performance music system––which I will describe more in detail in Chapter 5––several modalities of interaction emerged:

- **Human musician** performing gestures on a pressure-sensitive sensor surface with his fingers navigates in an interpolation control space

- **Physical objects** (weights) put on the sensor surface, pressing with a force proportional to their mass

- **Several objects** (e.g. steel spheres) moving on the sensor surface, colliding

- A **robotic sphere** moving on the surface according to the wirelessly obtained control instructions

- A **software agent** based on a neural network model trained on human gestures moves in a virtual space mapped to the the interpolation control space freely, or based on the input from the sensor (either from a human, or from a moving object)

- An animal, say, **hamster** walking on the sensor surface (actually, I have not really tried this one)

- A **complex virtual object** (physical simulation of a connected mass spring model) reacts to a sensor input, such a tapping on a surface, in turn modulating parameters of a sound generation engine in a non-linear fashion

It is obvious that probably not all of the agents in the examples would be considered autonomous, and also not at all of them are "artificial". But in order to be able to frame the different modes of interaction contributing in some way to the music generation, we need a broader theory capable of including other than human and software agents. And this is where Objec-Oriented Ontology comes handy: in view of the OOO the objects retain agency and thus we can view them as agents. So, in this context it makes sense to conclude that agents can be human, non-human such as animals, inorganic as mere physical objects, invisible –– software (computational, CGI), robots, or even hyperobjects such as the weather, in Timothy Morton's terminology (Morton 2013).

To sum up, in my understanding of the performance music systems I argue for a distributed agency (both performative and memetic) of interacting objects of different kind and hierarchies, but equal importance. They are acting on various scales and with a variable degree of autonomy. The heterogenous nature of these objects is inevitable by default: whether viewed as "cognitive assemblages" (Hayles 2016) or "structural couplings" among machines, performers and environments (Sanfilippo 2017), they create fluid constellations that constitute the basis of a performance with music systems.

The designers of generative and interactive music systems often mention the goal to create an autonomous agent. It is a historically old and tempting task to become the creator of some non-human "intelligence" or artificial life, but this approach has also its pitfalls, strange moments and uncanny valleys when machines are forced to imitate humans.[12] But as Collins states, there are also other practical concerns that it is not always the case that "the more intelligent, the better":

> *[...] for musical tasks which are dependent on a composer's aims, incorporating more advanced AI techniques will not necessarily make such compositions tractable, and there is danger they might over-complicate the situation.* (Collins 2006)

What I expect from a performance music system is a kind of symbiotic relationship, as opposed to a dominance of algorithms. Yet it might be often hard to achieve the former without also doing the

[12] Let us think of performance systems replicating the behaviour and sonic material of the human performers.

latter. To what extent do machines impose their behaviours on humans remains an open question. As I have noted, it has been common to project anthropomorphic concepts onto the machine objects, which is often manifested both in terminology and practical design approaches. But, said in the terms of evolutionary biology: if genes and memes use humans for their replication, machines and algorithms often make people to behave more like machines (think of gestures ”dictated” to us by the software, or the repetitive tasks connected with programming and debugging the code).

2.4 Computer Improvisation

> *Any specialized algorithm a composer might write to embody the generation of a piece—for example, one commencing with chaotic mathematics, and with preformed sections with different algorithms––could also be configured as an improvising platform, especially with added interface parameter controls for the real-time performers.* (Dean 2003, 56)

In his book *Hyperimprovisation* Roger T. Dean investigates the specific practice of interactive improvisation with computers (Ibid.). He gives an extensive overview of theories and examples of computer-mediated improvisation that has emerged as a new practice predominantly since the 1990s, so there is no need to cover them in detail here. I will therefore only highlight some of the most important features that I find relevant for building a music performance system which could be used in the improvisatory situations––whether in collective or solo performances.

It is evident that in the context of free improvisation, the expectations put on the performer are set rather high:

> *The improviser must effect real-time sensory and perceptual coding, optimal attention allocation, event interpretation, decision-making, prediction (of the actions of others), memory storage and recall, error correction, and movement control, and further, must integrate these processes into an optionally seamless set of musical statement that reflect both a personal perspective on musical organization and a capacity to affect listeners. Both speed and capacity constraints apply.* (Pressing in Dean 2003, xx)

In a musical improvisation, action and perception are tightly interlinked, involving different parts of performer's sensual and motoric circuits (Tubb 2010). An improviser is usually constantly switching between the performing and listening mode. John Priestley describes this process with a new term "poiesthesis": playing and listening are combined into a "single signifying practice" (Priestley 2014, 118). Of course, this could be true of any improvisation, but is to be taken into account even more when improvising with computers (Tahiroǧlu 2008).

However, in contrast to improvising with acoustic or electronic instruments with virtually fixed timbral properties, or collective improvisations with other musicians, the insertion of more or less transparent "mediation technologies[13]" intervening in the action-perception loop of a performer creates a universe of new possibilities. Most of all, algorithms let musicians shift their attention from details to high level abstration (Roads 2015, 341). But there is yet another difference between the improvisation with classical instruments and computers: the ability of the machines to generate a variety of new and surprising material. This feature could be very useful also in long term, given the fact that "*improvisation is usually distinguished as involving substantial fresh input to the work at the time of each and every performance*" (Nettl and Russell in Dean 2003, xiii). The possibilities of generative algorithms are vast, although they also carry potential dangers such as unexpected errors or "inappropriate" sounds. In case of the systems capable of relatively unpredictable sonic outputs, the importance of attention, feedback and immediate reaction is to be even more emphasized than when playing traditional instruments. It is a matter of performer's virtuosity which could be redefined in the new context as the ability to make musical use of such situations.

I will round up this section with a quotation accenting the spatial character of improvisation, which is considered as a gestural movement in a kind of imaginary space-time:

> *The imaginary space-time of improvisation is in itself a kernel structure for a compositional approach to improvisation since it creates a space for musical construction as if we were working out a compositional preconception. The dynamics of (hyper)gestures in the creative work of improvisation evokes a presence that is best described by Mihaly Csikszentmihalyi's flow concept and Keith Sawyer's extension to group flow. Flow transforms you in what they call "the zone." It is a presence that is atemporal, and we argue that this is precisely what needs imaginary space-time as localization. The improviser then lives a trancelike presence beyond physical space-time, a dreamlike state of inner balance. This higher level of consciousness is where so-called instant composition takes place.* (Mazzola et al. 2011, 247)

[13] "[...] in the domain of electronic and digital music production, there is a need for more *transparent* mediation technologies that create a feeling of nonmediation; as if the mediation technology *disappears* when it is used" (Bader 2018, 794).

2.5 Potential of Generative Strategies

The use of generative algorithms has a long tradition in computer music, especially in the non-real-time compositional domain. The range of the methods developed so far is very diverse and inspiring, including probabilistic models, chaotic equations, physical simulations, explorations of self-similarity and generative grammars, genetic evolutionary algorithms, machine learning or deep learning models of artificial neural networks. They can be used for rhythm and melody generation, timbral evolution or approximation, and many other purposes. With the overall increase of computing power and development of accessible software tools, the application of many of these strategies has been much more frequent also in the real-time context.

The key question is: how exactly can generative algorithms be useful and applicable in a performance system for musical improvisation? What strategy would be an efficient choice for a design of such a system? The conceptual appeal of the generative algorithms is big (Roads 2015), and the motivations for their application range from potentially surprising effects achieved by the use of stochastic methods and ordered complexity through romanticising of machine intelligence to a fascination with simulations of natural processes, such as the behaviours of inorganic or organic objects. My view here would be very close to the one foreshadowed by McCormack et al.:

> *It is not the indeterminacy that is of central importance here, but that the end product is something more than is specified in the instruction set […] as the process gives rise to outcomes that "outperform the designer's specifications."* (McCormack et al. 2012, 357)

One of the issues with the use of generative algorithms is that *"non-formal assumptions, preferences and subjective choices permeate the design and application of formal processes. This can make it difficult to assess the meaning of and significance of a generative approach." (Roads 2015, 339)*. However, I do not consider this as a disadvantage but rather as a ground for building experimental system, in Rheinberger's terminology.

2.6 System Feature Requirements

> *Creativity is often characterized as a combination of idea creation and idea selection. While this is a very simplistic model, these two processes usually form the core of more sophisticated multi-stage process models [...] We shall refer to these contrasting processes as "divergent" and "convergent," after Guilford (1967). (Tubb and Dixon 2014)*

With the increasing complexity of the system with its mapping and control layers, computer improvisers often have to combine **convergent** and **divergent** modes of thinking during the performance, which should be reflected in the interface design. Whilst the divergent mode allows for intuitive creative exploration, the convergent mode is oriented towards a specific goal, e.g. represented by a very concrete sonic idea. In computer improvisation it might be a good solution to to "outsource" some part of the divergent process to the computer, in Tubb's and Dixon's words. (R. Tubb and Dixon 2014). In this way it can be convenient to employ generative algorithms to search for new sonic "solutions", since they are able to quickly generate large amounts of data, *"but computers are rather bad at evaluating those data artistically: Listening to candidate sounds becomes the bottleneck." (R. Tubb and Dixon 2014).* In a real-time context this is then a good opportunity for human improvisatory skills to chime in.

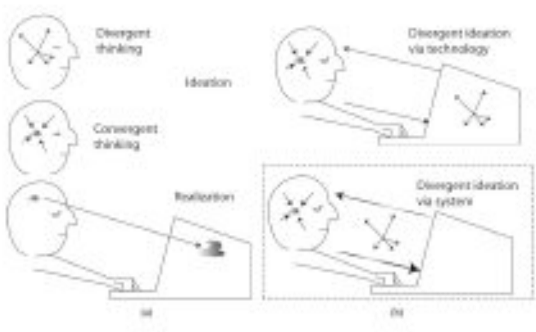

Fig. 4. a) Traditional view of the design of musical interfaces, b) Bi-directional interaction with technology where a feedback loop leads the the novel solutions and "transformational creativity" (Tubb and Dixon 2014).

In contrast with converging to the one and ”correct” solution, *“divergent ideation [...] would seem to require the ability to travel in any direction quickly, with varying levels of predictability.“* (Tubb and Dixon 2014, 24).

Since I fully agree with the view that the emphasis in free improvisation (and especially with algorithms) is on searching and exploration, this kind of practice poses specific demands on the musical performance systems (Mudd et al. 2014), apart from the general conditions such as robustness, performance stability or portability. Some of the most frequently demanded features computer improvisers usually mentioned as required from their tools are intuitive interaction (Kreković and Posćić 2015), expressiveness (Poupyrev et al. 2001), often along with with richness and sophistication of a traditional instrument. Based on the experience rooted in my own improvisatory practice combined with the outputs of the previous research in the field, I would sum up the considerations and design choices for a proposed system in several––sometimes closely linked––categories.

In computer improvisation, as opposed to composition, traversal of the space of possible solutions and their generation happens in real-time which means that speed is an important factor, as decisions about the next operations have to be made quickly. This has implications on the design considerations which I will deal with in a separate section.

In the following passage I will try to define my personal requirements for a music performance system using generative methods. They are based on my previous experience of improvisation with different kinds of electronic instruments, ranging from hardware devices (various analogue, digital and hybrid synthesizers, including modular synths, loopers, etc.) to software environments *(AudioMulch, Max* and *ppooll, SuperCollider, Ableton Live).* Some of these expectations are quite obvious or general, but nevertheless, they need to be mentioned as being part of the wishlist. In terms of real-time material generation, I consider following features to be important:[14]

> **Variety and variability:** The ability to generate a great variety of sound material that can be ”*actively investigated by the musician during the performance*” (Mudd et al. 2014, 126). Materials should be ‘malleable, capable of immediate adjustments” and available in “large quantities” (D. Wessel 1991, 347). In a live improvisatory situation this is crucial in order to have enough variety of material at disposal for the whole duration of a performance. The nature of the material though is mostly a matter of personal preference.

[14] I am covering the design requirements of the interface part in a dedicated section of the Chapter 5.

Complexity: The envisioned system should be capable of generating complex sonic output on different levels: timbral, rhythmic, and morphological. I would argue that large sonic complexity achievable in real-time is one of the most interesting benefits of employment of algorithms in music.

Fluidity: It should be possible to shape the generated sound morphologies in a continuous, fluid manner. I will focus on the aesthetic aspects of fluidity in the next section more in detail.

Unpredictability: A balance between deterministic and unpredictable sonic behaviour should be possible. Indeterminacy, unpredictability, instability or ambiguity are proven to contribute to the possibility of novel results, so they are often mentioned as an important features of music performance system for good reasons. This nature of complex systems could be emphasised and developed into “strategies for using the unexpected to advantage” in the performance, since error is a common source of musical ideas: *“unlike the well-trodden interface of an instrument in the hands or mouth of a player, the computer instrument can provide ambiguities, uncertainties, and variabilities galore.”* (Dean 2003, xvii).

Dynamics: To be possible to use the system in various unpredictable improvisatory situations, it should be capable of generating expressive sonic gestures in a wide dynamic range.

Several degrees of autonomy: To maintain a flexibility in different improvisatory situations, it should be possible to have selectable level of behaviour autonomy, ranging from self-evolving to human-controllable.

Modularity: Multiple sound generation engines and types of control options should be possible by design. It seems reasonable to keep the structure modular in the sense that the same “control core” of the system could be accessed from different entry points and various types of sound producing “objects” could be used (software or even hardware).

Flexible architecture: It should be possible to swap and change algorithms on the fly, in performance time (a semi-open structure).

In the next section I will examine the possibilities of sound material that could be generated by such a system.

2.7 Material Organisation

> *Mapping various fruits onto musical pitches works less well because fruit does not (in any ordinary way) constitute a continuum..* (Zbikowski 2002, 71)

Since the beginnings of electronic music, many thinkers and practitioners have insisted on the importance of the **continuum** as a concept for working with electronically generated sounds. For instance, the utopian vision of an access to the continuum of timbres was prevalent in the thoughts of electronic music pioneers, as noted in the writings of Karlheinz Stockhausen:

> *The discovery of the continuum between sound and noise [...] was extremely important because once such a continuum becomes available, you can compose it, you can organize it.* (Stockhausen 1989, 93)

Furthermore, it is not only Curtis Roads in his book *Composing Electronic Music*, who, elaborating on the ideas of Stockhausen, Gottfried Michael Koenig, John Cage and others, suggests to distinguish between different kinds of partly overlapping continuums that one can observe (perceptual) and compose for (parametric) on different time scales. Besides pitch-noise he also investigates the harmony-noise and the pitch-rhythm continuums (Roads 2015, 149). Of course, to take into account the real-world complexity of the sonic phenomena it is furthermore necessary to consider continuums with many more than one dimension. And although it is a psychological fact that for humans it is nearly impossible to think in more than three or maximally four dimensions simultaneously, I will later examine abstract geometrical models and practically useful tools that can help us orientate via projections into lower dimensional spaces.

Trevor Wishart is yet another composer who underlines the importance of understanding, exploring and controlling the whole musical continuum, imagined as non-lattice material:

> *The continuum is not an undifferentiated seamless fog opaque to human intellectual control but rather a wonderful new area for exploration provided we have tools to control the phenomenon [...] and the right conceptual categories to approach the material.* (Wishart 1996)

Moreover, stepping one level up, more examples can illustrate the fuzzy boundaries between different concepts that have been traditionally thought of as discrete:

> *Seeing music only through seemingly discrete concepts as 'style', 'work', 'instrument', 'melody', 'rhythm', 'notes' and indeed 'sounds' makes it difficult for us to imagine the configurations of variables outside of and between these categories for which there are no existing terms or concepts (yet)—configurations that might appear to be, say, part style, part instrument, or part melody, part timbre and part rhythm.* (Harper 2011, 58)

This is to mean that although we could imagine many of the musical variables as continuous, in the history of all music genres they have often faced some degrees of quantisation: notes with discrete rhythm values and precisely defined pitches, musical instruments with predetermined and basically fixed timbres, etc. To deconstruct the rigid nature of this kind of thinking, Adam Harper comes up with a more abstract proposal for a musical composition method, consisting of conceptual **dequantisation** of existing musical structures on various levels (not exclusively their timbral characteristics), followed by a subsequent requantisation:

> *If composers are truly to take advantage of music's utmost possibilities and set out into that vast sea of musical variability along a new route, they must dequantise what they know – break it down into its rawer, continuous variability – and then requantise by creating strange new configurations of musical variables from what they find.* (Ibid., 58)

This seemingly very general idea is partly visionary, as Harper's search for new music possibilities implies, and partly it reflects the tendencies of dequantisation on various levels of music making––from the level of sound objects within a particular piece to the fluidity of genres and fuzziness of their borders, from the perceptual to conceptual. It evokes an intriguing process of conceptual melting of the musical shapes into some kind of proto-substance, out of which new sonic structures could be built. Getting back to Stockhausen's idea, the question how to organize the available continuum, how to compose for it––and how to improvise with it–– is crucial and should be thought through on several levels.

2.7.1 Fluidity

The idea of the different continuums can be further expanded by looking at the fluid nature of the sound objects. It is said that Gottfried Michael Koenig *"rejected everything rigid in music: he was very sensitive to all stationary qualities of sound, to all sounds being unflexible objects." (Ungeheuer 1994, 27)*. Furthermore,

> *one hoped for (and categorically demanded) an unbroken continuum of all timbres; not only of all timbres, but the continuum between the timbre, stationary in itself, and the musical structure. The aim was the contoured, the fluctuating timbre. (Ungeheuer 1994)*

While in his composition *Terminus 2* the perceptual effect of gradual timbre changes was achieved by an admirable effort of manual cutting and splicing of large amounts of tape pieces, for modern computers it would be an easy task to do. But in relation to the new technological possibilities for sound generation, the physically-motivated metaphors of **flexibility, fluidity, fluctuation and liquidity** were often mentioned already decades ago as promising aesthetic concepts replacing the fixed, quantised structures. Liquid objects are unstable: they possess the ability to take any shape by continuously responding to the given conditions. In the theory of electronic music this is reminiscent of the Wishart's notion of **dynamic morphologies**: *"an object will be said to have a dynamic morphology if all, or most, of its properties are in a state of change" (Wishart 1996, 93)*. Another option is to describe these behaviours as **flowing streams**, where, again, the *"streaming mesostructures seem to flow rapidly like liquids"* (Roads 2015, 310). Dennis Smalley, among others, also mentions processes of motion such as **flowing** or **floating** (Smalley 1997, 117). Already in his famous lecture in 1936, Varèse formulated similar visions evoking fluid matter:

> *When new instruments will allow me to write music as I conceive it, taking the place of the linear counterpoint, the movement of sound-masses, of shifting planes, will be clearly perceived. When these sound-masses collide the phenomena of penetration or repulsion will seem to occur. Certain transmutations taking place on certain planes will seem to be projected onto other planes, moving at different speeds and at different angles. There will no longer be the old conception of melody or interplay of melodies. The entire work will be a melodic totality. The entire work will flow as a river flows. (*Edgard Varèse in Schwartz, Childs, and Fox 1998)

It is precisely this fluid character of sound objects that it is so tempting to get under real-time control: to be able to generate and shape the non-stationary morphologies on the fly.

2.7.2 Motion

When trying to classify the modalities of dynamic morphologies, several types of continuous changes or fluctuations of the sonic material can be identified. From an analytical point of view, it is possible to define terms describing abstract spatiotemporal behaviours in the perceptual space, for which then there is a task of finding and designing corresponding correlates in the parametric space of the sound generating algorithms.[15]

For instance, Smalley in his theory of spectromorphology, devoted to the exploration of spatiotemporal sonic shapes in the perceptual space, describes the processes of motion and growth (Smalley 1997, 115) and suggests a classification illustrated in Figures 5 and 6.

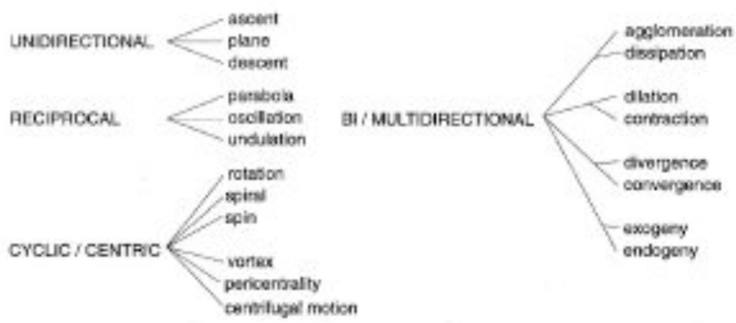

Fig. 5. Motion and growth processes.(Smalley 1997, 115).

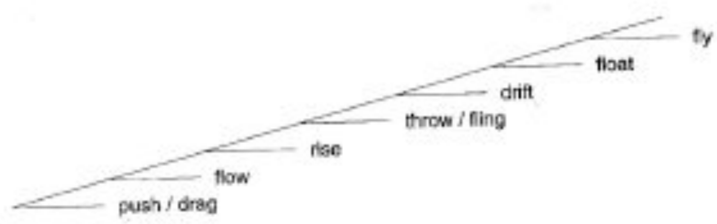

Fig. 6. Seven characteristic motions.(Smalley 1997, 115).

[15] The characteristics of each of these spaces will be explained in the next chapter.

For the purpose of the design of a real-time performance system, I am especially interested in the exploration of tension and ambiguity resulting from the movement between the clearly defined states. In respect to this, several types of motion are exceptionally interesting:

– **Oscillation**, when the qualities of a sound object periodically or non-periodically evolve around a perceptual centre that could be described as an attractor in a n-dimensional feature space.

– **Mutation,** meaning changing the identity of sounds through evolution in time (Roads 2015, 287).

– **Drifting.** Continuous goal-less mutations of selected attributes can be described as drifting of the morphologies in the feature space.

– **Morphing.** Specific combinations of parameters, whether generated automatically or by manual fine-tuning, result in recognisable spatiotemporal gestalts. The process of morphing can be perceived as a gradual change between the two of these. The concepts of betweenness, ambiguity, similarity and variation can be explored by movement in the n-dimensional feature space.

For all these options it is further possible to define second order properties such as the speed and amount of deviation or change, based on the categories of proximity or distance in the feature space. In contrast to oscillation, mutation and drifting behaviours, morphing is a goal-oriented process that could be represented by a specified trajectory and “destination”.

As mentioned previously, the idea to access various ”in-between” configurations of variables and the possibility of gradual transitions in multiple dimensions are certainly intriguing when thinking about real-time applications. Such kinds of possibilities of transitions between different ”states“ were already studied by Iannis Xenakis, who suggested to use stochastic methods for their control:

> *Passages from a discontinuous state to a continuous state are controllable with the aid of probability theory. For some time now I have been conducting these fascinating experiments in instrumental works; but the mathematical character of this music has frightened musicians and has made the approach especially difficult.* (Xenakis 1992)

But again, it has to be underlined that this idea goes beyond timbre as it can be applied to the temporal development in the continuum(s) with many dimensions (rhythm, pitch, etc). Creation of such spatiotemporal sonic movements is essentially what Wishart means by a "musical gesture", denoting a multidimensional articulation of sound in parameter space over time (Wishart 1996, 43).

One could, of course, imagine some more detailed typologies of dynamic morphologies, which is undoubtedly a tempting idea. Wishart tries to sketch some possible directions, excited by the "catastrophe theory"––a branch of bifurcation theory in the study of dynamical systems, often referred to as a theory of discontinuous change (Stewart 1976). While many of the shapes described by the theory can be certainly appealing for the design of algorithmic behaviours, Wishart himself finally ends up with a somewhat reserved attitude towards the formalist urge for universal categorisations, mainly for the reason of difficult applicability of the theory's finding in higher-dimensional spaces.

2.7.3 Time Scale Operations

The writings about form in electronic music usually evolve around the ways of temporal organisation of the sonic material on three time scale levels: micro (milliseconds-seconds), meso (seconds-tens of seconds) and macro (tens of seconds-tens of minutes). In Roads' investigation of the sound organisation on the meso and macro time scales I find some of the concepts very relevant and especially challenging to implement in a realtime improvisatory situation (Roads 2015).

An ambition to strictly impose the top-down logical rules on the organisation of the sound material has not only often proven as a failing compositional approach (Roads 2015), but moreover, it is exactly the opposite which forms the very essence of improvisation. This can be, of course, also applied to an interaction with generative processes.

Improvisation with algorithms has the benefits of joining the **intuitive** and **systematic approach**es, which happens to be often the preferred way of compositional working, even with non-real-time methods (Xenakis, Stockhausen). This hybrid formal/informal method is *"combining computational power of algorithmic control with the magical influence of heuristics" (Roads 2015, 351).* In a heuristic, experience-based and exploratory mode of operation the performer constantly searches in the space of momentary decisions about the next directions, which takes place in a constant dialogue with the stream of material resulting from the machine agency. The "division of labour" and responsibility for the sonic process between the performer and the generative algorithms always pose an interesting task that can be experimented with in different ways. In general, stochastic methods or multiagent modelling in the generation and shaping of sound objects tend to be very useful for the creation of a variety of sonic material especially on lower timescales (micro to meso level). However, when considering sonic development on medium to larger scales in performance time, the bottom-up approach often fails to address the creation of perceptually meaningful narrative structures (Roads 2015). Since they are highly context-sensitive, it is hard or impossible to program them as rules. Therefore, this is the point where human agency can be important.

In the context of an algorithmic improvisation it can be said that the **macro form** emerges from real-time organisation of instantly generated sound objects with different behaviours, grouped into shapes that Roads calls mesostructures. It is fascinating how reminiscent this is of Edgard Varèse's description of his compositional process, even before the digital era:

> *There is an idea, the basis of an internal structure, expanded and split into different shapes or groups of sound constantly changing in shape, direction, and speed, attracted and repulsed by various forces. The form of the work is the consequence of this interaction. (*Varèse in Schwartz, Childs, and Fox 1998)

Based on their length in time, these sound objects could be categorized as short rhythmic pulses, phrases or longer textures, but basically we are still operating on the meso time scale. However, it is precisely the instability and mutability of the sonic objects in various dimensions––including the temporal––that challenges this division. Since I am interested in deviations and morphing of one gestalt into the other, the term **fluctuating mesostructures** seems highly appropriate here.

The pitfalls of both the top-down and bottom-up approach can be avoided by an––again risky––intuitive method with the properties of an exploratory and open-ended process that Roads calls multiscale planning. It can be likened to

> *solving an n-dimensional jigsaw puzzle, where each piece in the puzzle is a sound object with a potentially unique morphology. How the pieces will ultimately fit together is not evident at the beginning. As the composer assembles the puzzle, certain objects appear to be natural matches. They fit in sequence or in parallel. Other objects seem out of place.* (Roads 2015, 303)

Furthermore, I find the dynamic notion of the "partial systems" is of a special relevance:

> *Multiscale organization can be likened to a heterarchy of partial systems that come into and go out of being. It can employ generative processes but reserves the right for the composer to interact, intervene, edit and transform at any time.* (Ibid., 317)

In the proposed version, multiscale planning refers to the compositional process that happens outside of performance time. But for the real-time aims I would propose to extend this concept by introducing the term **multiscale improvisation.** Since, as the author notes, such a process is *"difficult to describe precisely in words or in computer programs" (Roads 2015, 304)*, it is exactly where and how algorithms and human performers can ideally complement each other.

2.7.4 Shapes in Multidimensional Spaces

The properties of dynamic morphologies can be described by both their temporal development and other features understood as "spatial" in an abstract sense. This refers to the idea of the attributes of sonic objects represented in a multidimensional mathematical space. The spatial metaphor is very common, as I will demonstrate futher, because it has considerable cognitive advantages. As illustrated e.g. by Roads in the Figure 7, the morphology of a sound object evolving in time can be viewed as an "*n*-dimensional space of envelopes and constants":

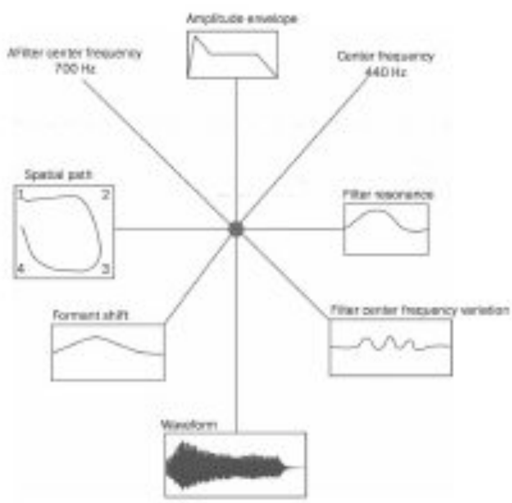

Fig. 7. Morphology of a time-evolving sound object (Ibid., 303).

A step towards spatial organization makes the properties of dynamic morphologies conceptually accessible, in that the universe of possible sonic material can be organised in both cognitively and computationally meaningful manner. One of the options is to further organise the properties of the material on any time scale according to the axes based on higher-level perceptually-based oppositions (Roads), or––as their equivalent in the control space––polarities (Gunnarsson 2012, 40). These oppositions can be related to numerous "spatiotemporal" (in the abstract sense) qualities of the sound objects and their amount can be arbitrarily big. For instance, rhythmic oppositions according to Roads could look like this: constant rate – variable rate (rubato), regular (no intermittencies) – regular intermittencies stipulated as a burst ratio, sparse – dense, etc. (Roads 2015, 184). In the *Handbook of Systematic Musicology* Rolf Inge Godøy compiles another possible list, related to texture "as situated in a multidimensional feature space": dense – spread, thick – thin, synchronous onsets – asynchronous onsets, short tones – long tones, many sustained tones – few or no sustained tones, wet – dry, little or no melodic movement – much melodic movement, small intervals in melodic lines – large intervals in melodic lines, etc. (Bader 2018, 771).

Interestingly, the oppositions describing sound morphologies in the perceptual space can have their counterparts in the computational control space, influencing the algorithms generating sound. This is the case of the objects controlling generative processes in Bjarni Gunnarsson's environment for algorithmic composition EPOC:[16]

> *An object is in control of a sound process and can also be subject to other structural processes or direct interactivity from somebody using the environment. This user can in various ways create configurations for how objects will behave and interact. However, once the system is running, the main way of communication will be provided by the object and its polarities. The polarities provide a common language that all objects understand and respond to.* (Gunnarsson 2012, 40)

For each of the possible EPOC's objects representing various generative algorithms, settings for nine polarities in following categories can be defined in the range from 0 to 1: speed (rate of activity within the process), density ("mass per unit volume in the sense of filling the acoustic space", entropy (measure of uncertainty), position ("process offset"), amplitude, frequency (low, middle, high), surface (representing timbre attributes from soft to rough), color (related to spectral density) and location (in the perceptual field).

[16] This was inspired partly by Koenig's compositional system used for the piece Project one. (Ibid.)

It is evident that a powerful musical performance system can be designed to enable a dynamic exploration of such oppositions or polarities. Although the effect of their settings on the processes can be explored individually (e.g. with one-dimensional controllers), technically they form a nine-dimensional control space for a higher-order control of complex algorithmic structures. As can be observed, there are certain correlations between the control parameter settings and the perceptual qualities, but the correspondences are not straight forward. In case a performer would like to control such system in a real-time context and in a more embodied or intimate way than by a series of onedimensional mappings of the input to control parameters, issues of possible geometries and connections between different kinds of abstract spaces would start to appear.

When Roads writes that "*from a compositional point of view, music is an n-dimensional design space in the sense that there are no intrinsic limits on the type and number of independent parameters that a composer can conceive and manipulate.*" (Roads 2015, 289), it is reminiscent of Wiggins' observations on the cognitive category of a conceptual space:

> *A collection of spaces may be co-originally superimposed to create a manifold of dimensions which are capable of describing, in principle, any structure. (Wiggins 2018b, 15)*

This is a very abstract but also an intriguing idea, although such a manifold is an incredibly large space and when it comes to practical applications, it would always have to be reduced or divided to much smaller subspaces.

Coming back to the Creative Music Framework of Wiggins and Rheinbergers' notion of experimental systems from the beginning of this chapter, my goal could be described as to create a music performance system where a subset of the universe of possible concepts in particular constellation can be traversed and the solutions evaluated by the performer on the fly. The structure of this process was essentially depicted earlier in the Figure 2, but it should be happening in real time.

The spatial perspective opens some interesting questions that are worth deeper investigation. How are these spaces related or aligned? What other mathematical, computational or cognitive spaces can exist or are deployed as tools in relation to electronic music? And, last but not least, how can they be effectively connected in the design of music performance systems and how can be worked with multidimensional hierarchies in real-time? Mathematician René Thom makes an accurate point here: "*The first objective is to characterize a phenomenon as shape, as a spatial shape. To understand means first of all to geometrise.*" (Godøy 2018, 771). In the next chapter, I will provide an overview of the various concepts of space in relation to electronic music. I am convinced that the understanding of different notions is very relevant to thinking of of a digital music performance system with many computational parameters, as it unfolds from the idea of specific connections between several types of abstract spaces––such as input conceptual search space, feature space, interpolation control space, algorithmic parameter space and the perceptual sonic space.

3 Spaces in Music

3.1 Ideas of Space

> *Correlating musical pitches with vertically oriented, two-dimensional space, for instance, leads naturally to an imaginary world in which pitches become things that move through space: the successive notes of a scale gradually descend and ascend; in other passages, some notes leap, while still others fall. Within this imaginary world, each traversal of space has a specific and unmistakable sound — that is, descent sounds one way, ascent another. And this is not something limited to text painting of the sort demonstrated by Palestrina, as any number of cartoon soundtracks confirm.* (Zbikowski 2002, 65)

Since the early days of electronic music, various concepts of space in association with sound have been popping up as used by music theorists, composers and musicians with both analytical or practical goals. The theoretical and practical interests have been very much, and logically, interconnected: the urge of understanding of the sonic phenomena has been also a path leading to the discoveries of new compositional approaches. Since the notion of space belongs to one of the most basic cognitive categories, the prevalence of spatial thinking on the various levels in the music discourse does not come as a big surprise. Even more so, if we consider the well known fact that the roots of geometrical thinking are even much older than the history of electronic music (Mazzola 2003). If we think of musical scores, vertical and horizontal pitch organisation, harmony and counterpoint with their symmetries, circular rhythm structures and the like,[17] much has been written of the spatial organization of the musical material on a conceptual level. It is, nevertheless, interesting to see that also in the very recent years we could have observed a rising interest in the geometrical approaches to the musical thinking (Tymoczko 2011). It is obvious that this tendency, likely driven by the increasingly complex and data-based nature of the today's world, is not only

[17] "In music cognition research, as elsewhere in psychology, success in understanding perceptual similarity has been achieved by modelling the behaviour of musical percepts and structures by means of low-dimensional geometrical spaces, such as that of pitch, and numerous related proposals (implicitly or explicity geometrical) exist to explicate Western tonal harmony. A geometrical space has even been proposed to capture the full complexity of musical metre and rhythm" (Wiggins 2018b).

present in music. We can notice it in various other areas, with the applications ranging from machine learning and robotics to the emergence of new analytical tools for making meaning of big data, or even whole disciplines such as spatial computing(Bigo, Spicher, and Michel 2010). Joel Ryan observes here a convergence of simulation and imagination:

> *Mathematical ideas themselves are more easily communicated and elucidated via graphic simulation. In fact the very nature of scientific rhetoric itself is changing so that proof "by construction" is once more mathematically valid. These systems have drawn so close to the "language of science" that we now trust in simulations to search for the solution to both pure theoretical and more practical as in cosmology, meteorology and geophysics. This convergence of simulation and imagination will probably be as empowering for our time as was the discovery that the syntax of algebra enables extension of the idea it represents.* (Ryan 2002)

Despite the spread of this phenomenon, it is hard to find a comprehensive theoretical underpinning of electronic music that would take into account different use cases of the concept of space in all their different meanings and applications. A theoretical framework like this would inevitably have to combine the knowledge of various disciplines: music theory, psychology, cognitive science and theory of creativity, data science and human-computer interaction, user experience design, music information retrieval and musical instrument design as well as practical music making including composition and improvisation. In all these fields, spatial concepts have contributed to the development of theoretical and practical applications. Even if a comprehensive theory might not be needed, I felt the need to summarise all the different concepts and to bring them to a common ground.

We could mention Denis Smalley's seminal contribution to the field of electroacoustic music research, with the enumerations of the use of the term space in different contexts: from *perspectival space* and *spectral space* through *gestural space, ensemble space, performed space,* to *microphone space* and many more notions (Smalley 2007). Most of these spaces belong to the perceptual descriptions, or they fall somewhere in between physical and psychological, with a different degree of abstraction. Geraint Wiggins, for instance, came up with a framework for creative systems in music, where he suggests various ways of searching in the conceptual space to generate new solutions (Forth, Wiggins, and McLean 2010). We also know are many examples of generative music systems with self-organizing swarms of agents, or other artificial life forms and ecosystems 'living' in *virtual spaces* (Miranda 2011, Herber 2010). Furthermore, recently there has been a new wave of the 'artificially intelligent' digital musical instruments that use the method of moving in the so-called *latent space* of a neural network to produce variations of musical material (Roberts, Engel, Oore, et al. 2018). The game developers often mention the *spaces of possibility* (Kanaga 2018), whilst composers mention the concepts of a *sonic space* (Wishart 1996) or *music space* (Harper 2011).

When tracing the relationship of sound and various meanings of the term space, we can identify not only the different levels of abstraction in the concepts of space and their various accuracy or amount of detail, but also their various ontological statuses. How do these spaces exist? How "real" are they and how can they potentially be aligned together? Moreover, are they static structures or can they change their geometry over time? These questions are not easy to solve and maybe to there is no final and satisfying answer. But it is clear that whereas some terms relate to perceptual phenomena studied by psychology, others belong to the more abstract mathematical representations. Several of them we could see as descriptive models, whilst others have explanatory or even predictive capabilities. From the basic definitions, we could move on to the practicalities of how these ideas of space could be used in a creative musical work.

Frederico Macedo identifies five different notions of space often used in the research related to sound and music (Macedo 2014, 64):

> *[1] space as metaphor.* This includes all the spatial images describing *"abstract concepts or perceptual experiences associated with sound and music, not necessarily related to the spatial perception of sound".*
>
> *[2] space as acoustic space,* representing the physical space with its properties that have an acoustic impact on sound (this includes reflection, diffraction and resonance)
>
> *[3] space as sound spatialisation*—the auditory field, containing information about positioning and movement of sound sources perceptual field
>
> *[4] space as reference*—related to the ability of sound to refer to specific places, and
>
> *[5] space as location,* referring to the physical presence and experience of being at a specific place, *"including its cultural, historical and environmental implications".*

The idea of a space as metaphor is meant to include all possible mathematical objects related to the sound phenomena. E.g. the vector space models have proven to be very useful epistemic tools and as such they are often used for scientific analysis or as creative models for music making.

I would regard this categorisation as useful, although rather descriptive than forming a unified cognitive theory of the spatial concepts in music. There are also details that might need more clarification. We could ask, e.g. where the spatial simulations such as the virtual reality including some sound elements belong––probably to several categories such as 2, 3 and even 4 and 5, since they are a quite complex phenomenon. Another categorisation problem arises with the mathematical computational spaces that are used as constructive tools, such as n-dimensional data representations.

While it is true that not all the invented spatial metaphors or models have[18] proven to be effective over the years,[19] some concepts did turn out to be indeed useful. One of the fruitful ideas, for example, is to see data structures representing music as topological spaces (Giavitto and Michel 2002). This enables us to do various geometrical computational operations with them, which opens up interesting musical possibilities.

Several strategies offer us to use geometrical representations that reduce the often multidimensional information in a more or less sophisticated ways to make it meaningful and accessible for practical use. Since one of the deepest problems in cognitive science is the way people make sense of the vast amount of information surrounding them (Hofstadter 1995, 169), it is interesting to observe how the inherently spatial nature of human perception led to the formation of higher-level and more abstract concepts useful for building instruments, understanding sound phenomena and creating music (Giavitto 2015).

When studying the use of geometrical concepts in music, it can be seen that many of the––even very recent––geometrical ideas have been applied to modelling lattice-based structural relationships (Tymoczko 2011, Blondeau 2017). Since the main research interests of electronic music escape this paradigm in favour of the widely open sound continuum, in the following section I am briefly investigating some of the most relevant notions of spatial representations and concepts that are related to the domain of music as organised sound and seem to be relevant for the construction of new musical interfaces. This catalogue is meant to be by no means exhaustive but it can nevertheless provide illustrations of the different representations and their practical applications.

[18] Scientist in the engineering sciences build models for thepurposes of imagining and reasoning about how to improve the performance ofthe devices, processes or materials of interest. These models involve imaginableproperties and processes, and they incorporate measurable physical variables andparameters (e.g. in the case of chemical engineering chemical concentrations, flowrates, temperature, and properties of materials such as diffusion, viscosity, density). Often, these models also incorporate dimensions of typical configurationsof certain devices. (Boon and Knuuttila 2009, 696).

[19] Such as the way Thomas Clifton tried the mapping of melodies to surfaces (Macedo 2015, 225).

3.2 Music Space

Throughout the 20th Century's music history, the idea of a sound-space originally conceived by John Cage as a sum of all possible states of several basic sound-related attributes such as pitch, volume and timbre,[20] has proven to be a concept that has since inspired many musical thinkers (Cage 1961, 9). As an important contribution to the emerging electronic music discourse, Cage described the role of technology (such as tape manipulations) as a tool that had enabled the composers to expand the "space of possible sounds" (Cage 1961, 9). Half a century later, Adam Harper refers to this idea when further elaborating on the spatial concept by proposing a general framework for music making. He describes composition as a general process of working with many variables (Harper 2011, 17). The variables can be seen as either independent or linked together, and besides determining the particular attributes of sound, they can also be used to describe relationships between other variables. In addition to that, they can refer to the properties of time-based structures, repetitions, or any other quantifiable aspects of musical works. Moreover, these variables do not only have to relate to the qualities of sound or structural elements of music, but they could take into account the relations of sound processes to the "outside world", such as those of the physical space where a particular music performance should be held. Music can thus be understood as a process of (consciously or unconsciously) dealing with very large and complex sets of variables. This view actually underlines the possible variability of the musical work, one that comes before the sounds themselves. Such a broad definition has an advantage in that it can offer a "parametrised" concept of music making which can include e.g. musical instruments, but also whole styles as well as particular works as configurations of variables. Harper further continues by defining a music space as an imaginary multidimensional totality of possible configurations. The temporal dimension is then added by a metaphor of travelling through this space:

> *In this part, we'll refer to that variability of music as a whole as music space. I've already described music space as a 'sea' and suggested that specific variables can be thought of as 'paths' or 'routes' through it. Calling this total variability a 'space' continues the analogy, imagining musical possibility in a similarly geometrical or geographical way. We'll also refer to limited configurations of musical variables together with the constraints (i.e. quantisations, ranges) imposed on the values of those variables as musical objects. This is both for the sake of convenience and because those*

[20] In Cage's view timbre has been accounted for as one dimension, but it has since proven to be more of a complex 'umbrella' category, which could in fact consist of many more individual variables.

limited configurations are the objects that make up music, where terms like 'sounds', 'melodies', 'pieces of music' and 'instruments' are, as we've seen, either too vague or too specific as concepts. Music space, then, is the continuous space or continuum formed by all musical objects. (Harper 2011, 89)

For the sake of clarity it is worth noting that notion of *music space* or *musical space* is one of the terms overloaded with meanings and it has been used by many authors in different ways (Emmerson 1998).

The metaphorical concept of the multitude of variables constituting a musical space will be used as a general foundation for further development, where many other spatial representations related to music can be thought of as subsets of this, very broadly defined space. Later on, I will investigate the suggested properties and possible operations in this space, such as continuity, quantisation, dequantisation, and others.

3.3 Timbre Space, Sonic and Sound Space

Defining timbre is a hazardous operation. (Smalley 1994)

The term "timbre", depicting the "colour", or perceptual quality of sound, has been often viewed as a multidimensional property, although a more precise definition has been still posing a theoretical problem: "*In discussing the attributes of complex tones, Licklider (1951) concluded that "until careful scientific work has been done on the subject, it can hardly be possible to say more about timbre than that it is a 'multidimensional' dimension."* (Grey 1977, 1270). The attempts to describe this phenomenon led to the introduction of a *timbre space* as a representation *"derived from the timbre dissimilarities among a set of complex tones"* (Seago 2008).

The timbre-space representation is a powerful psychological model that allows predictions to be made about timbre perception in situations beyond those used to derive the model in the first place. Timbre intervals, for example, can be conceived as vectors within the space of common dimensions. Timbre *space also makes at least qualitative predictions about the magnitude of timbre differences that will provoke auditory stream segregation.* (McAdams and Giordano 2012, 1:78)

On the one hand side, it aims to represent an imaginary space of all possible timbres. But, on the other, to achieve a human-comprehensible depiction it is necessary to reduce somehow the number of dimensions. The issues with this approach are mostly based on the fact that the perceptual qualities of "sound colour" do not seem to correspond directly to any exactly measurable attributes. But, several authors tried to represent this kind of space in two or three dimensions by technological means. One of the early visualisation attempts by John Grey in his classic research of a technique called multidimensional perceptual scaling was based on a computer-based clustering of perceptual data obtained from a group of musically educated listeners asked for the similarity judgements between the pairs of sounds. The resulting data were projected into a 3D coordinate space as locations representing the timbres of digitised acoustic instrument sounds, as depicted in Figure 8.

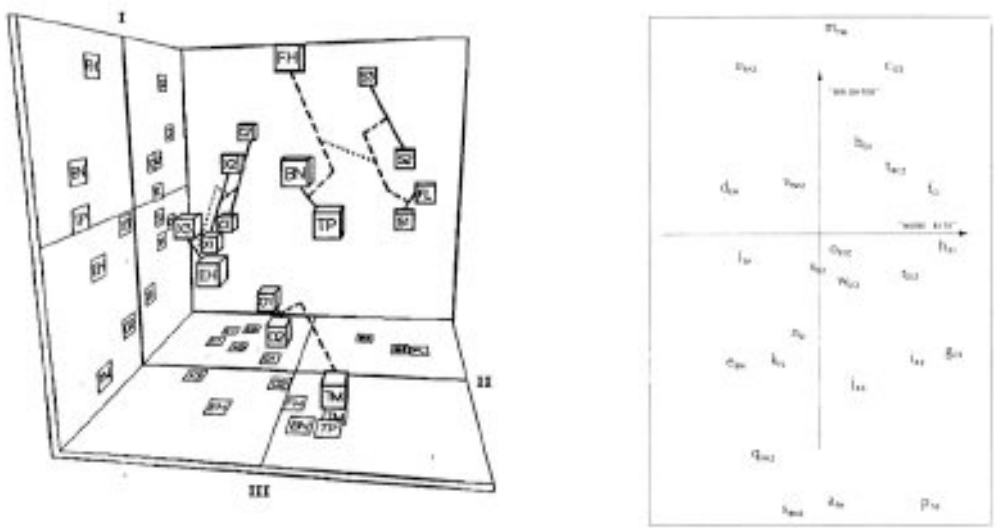

Fig. 8. Left – abbreviations: O1, O2 = oboes, FH = French horn, BN = bassoon, C1, C2 = clarinets, FL = flute, X1 X2, X3 = saxophones, TP = trumpet, EH = English horn, S1, S2, S3 = strings, FHZ = modified FH withspectral envelope, BNZ = modified BN with FH spectral envelope, S1Z = modified S1 with S2 spectral envelope, S2Z = modified S2 with S1 spectral envelope, TMZ = modified TM with TP spectral envelope, BCZ = modified C2 with O1 spectral envelope, O1Z = modified O1 with C2 spectral envelope (Grey 1977, 1270). Right: (D. L. Wessel 1979, 49–51).

David Wessel (right picture in the Figure 8) proposed a 2-dimensional representation of the orchestral sound timbres instead. The rationale behind the alignment on a surface was his goal to create control systems for music production based on perceptual representations with the available sensors, such as the graphic tablet. Based on the particular location in the timbre space, it would be possible to access the sound with the timbre represented by the coordinates[21] (D. L. Wessel 1979, 49–51).

However, it should be said that due to a simplified representation, these projections do not take into account time as an important dimension in which the sounds unfold and change. We can think about them as representing only slides of a more-dimensional phenomenon.

Trevor Wishart mentions yet another concept. In his view, what is called a *sonic space* is defined by three dimensions: pitch continuum, noise colouration and *timbre space*. It is interesting to observe that whereas one of the axis (the pitch continuum) allows for discretization of its values, the noise colouration seems to be missing this choice (Macedo 2015). Although Wishart investigates the topology of the timbral space and the possibilities of its navigation, he resists a more precise formalisation of its parameters (Wishart 1996, 82). But along with theorising about a uniform and potentially infinite timbral space with fuzzy borders, he also brings in a useful idea of its subspace, describing the sonic possibilities of a particular instrument.

As being connected to timbre as well, Denis Smalley uses a concept of a *spectral space* (Smalley 2007, 36), referring to the spatial depiction of frequency distributions containing areas "occupied by sounds" and their components. It is possible to create a spectral analysis from a sound and visualise, or even edit the data as a 3- or more-dimensional sonogram that also includes the time axis. Although this approach can give us an objective representation and direct manipulation of the sound components, the correlation between the perceptual timbral qualities and the representations of the sound spectra such as that shown in Figure 9 is not a straight forward one. Therefore, to be able to synthesise sounds based on the timbral adjectives, we need to use some methods of perceptually controlled sound synthesis.

[21] This idea could be tested in a real-time setting with the graphics terminal of the computer at IRCAM in 1976, controlling an oscillator bank.

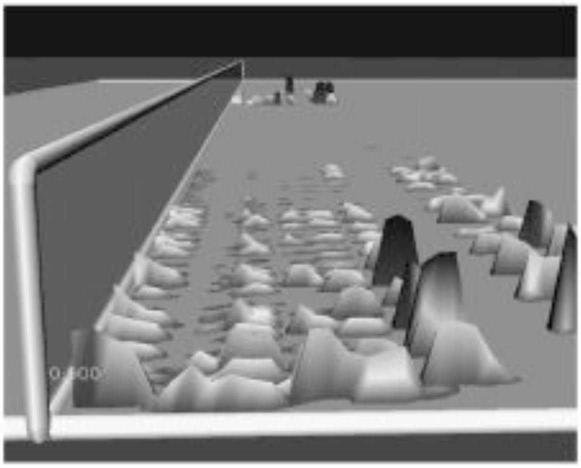

Fig. 9. OpenSoundEdit software illustrating the time-varying sound spectra (Chaudhary, Freed, and Rowe, n.d.).

Based on the method of spectral analysis, it is also possible to construct projections of the timbre space to spaces of lower dimensions (two or three) by using various sound descriptors of choice as its axes. Such an approach can be found in control interfaces for the so-called corpus-based synthesis, with sound samples of different properties distributed in a “playable” space[22]. Since several analytical descriptors such as loudness, pitch or spectral centroid correlate with perceptual notions of amplitude, frequency and brightness, it is possible to access the desired sounds intuitively. With this flexible visualisation technique, the actual view in the Figure 10 is, again, an actual 3D slice of a larger n-dimensional space:

[22] In this case, each of the 12 descriptors can be selected as one of the axes (Loudness, Pitch, Note Number, Periodicity, Spectral Flatness, Spectral Centroid, High Frequency Energy, Mid Frequency Energy, High Frequency Content, AC1, Energy, Label). http://madeby.userstudio.fr/swirls-for-catart/

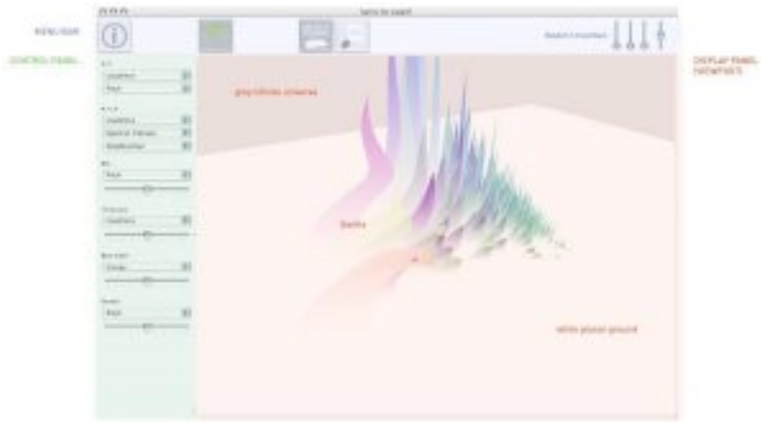

Fig. 10. Swirls for CataRT (with selectable descriptor axes on the left).

We can say that the *timbre space*, *sound space* and *sonic space* with their subspaces such as Wishart's *instrument space* are multidimensional metaphorical notations of the sound characteristics derived from the perceptual experience. An investigation of how close the machine-analysed sound descriptions would be in comparison with their positions in timbre space based on the perceptual scaling technique showed that, as expected, there were some correlations (Toiviainen, Kaipainen, and Louhivuori 1995). The early techniques of spatial representations of timbre have led to the construction of real-time timbre exploration and editing tools such as Spear[23] or Loris,[24] or the method of audio mosaicing, where one timbre space is being mapped onto a different one (Schwarz 2012).

As a frequently recurring topic, the possibility of continuous morphing between the timbres seems to be a long-term-intriguing task in the electronic music creation, which I will cover extensively later.

[23] http://www.klingbeil.com/spear
[24] http://www.cerlsoundgroup.org/Loris

3.4 Parameter Space

> *Each type of cuisine can be defined by a central parameter that determines the other ones. For example, in Mexican cuisine spicy is prioritized over all.* (Egido 2015)

The electronic sound creation requires the musicians to deal with the sets of parameters that determine the settings of the signal generation or processing circuits or algorithms. The so-called parametric thinking has been gaining an increasing importance over the past years in many creative disciplines including media art, design and architecture, which can be partly attributed to the ubiquitous use of technologies. In the history of Western classical music though it has been prevalent since the spread of the serial composition technique after the World War II that aimed to get a total control over the musical parameters.

The amount of parameters in the sound production tools often increases with the level of sophistication of the software or hardware architecture, as well as with the complexity of the desired sonic results. One of the useful geometrical tools that can help us imagine the possibilities of such an instrument, engine or setup is the concept of a *parameter space.* It is based on the mathematical idea of a topological vector space representing the collection of all possible configurations of the parameter settings. These configurations can be described as vectors or points positioned in a multidimensional abstract space. Based on the nature of the parameters, this space can be considered continuous, or it can contain ruptures that prohibit smooth transitions.

One of the most important questions when considering the sound generation possibilities addresses the relationship between the parameter space as a concept rooted in engineering, and the perceptually defined spaces, such as the––also multidimensional––timbre space or sound space. As the intuition suggests, these spaces are correlated, but their alignment is complicated and strongly depends on the sound producing architecture.

> *We assert that the physical parameter space of a musical instrument has a strong bearing over the perceptual space that is invoked in the listener. In other words, the listener in part perceives a sound as a movement in the parameter space that invoked it.* (Forth, Mclean, and Wiggins 2008)

Nevertheless, it is important to mention that even though many researchers have explored the mappings between the parameter space of the sound synthesis and the timbre space, the space of parameters can extend way beyond that of the synthesis, basically defining any possible process or behaviour. Parameter configurations can be also used to construct sequences or functions determining the temporal development, level of interactivity or any other behaviour. Since the potential number of parameters of a sound generation process could be immense, we could now question the relationship of the parameter space and the music space as proposed by Adam Harper. It turns out that although they are by no means identical, we could say that the parameter space in the technical sense is a subset of a much larger musical space that the composer has to consider. Similarly to the variables, we can consider also parameters with different degrees of interdependence or discontinuities which makes the topology of the space a more complex issue.

One of the other biggest problems in the electronic instrument design addressed by diverse, the so-called mapping strategies discussed in the following sections, is the disproportion between the number of dimensions of the control space versus the much larger amount of sound generation parameters. Departing from this point, most of the tools presented further try to address this issue to achieve an intuitive navigation of the parameter spaces.

3.5 Control Space

> *In applying their awareness of multidimensional timbral features, composers, aided and abetted by technology, are often absorbed with concepts, methods and techniques: the listener's apprehension of timbral values cannot simply be equated with the launching of multidimensional attributes by the composer.* (Smalley 1994)

As noted earlier, there is often a disproportion between the huge dimensionality of the possible sound generation parameter spaces and the timbre spaces representing the perceptual qualities of the resulting sounds, which poses in a challenge in the design of tools and systems for the practical use. Even typical commercially available hardware or software synthesizers can have hundreds of parameters, not mentioning more advanced ways of sound generation, where the size of the parameter space is limited only by the computational resources. The *control space* in general corresponds to the multidimensional space defined by the accessible controls of the parameters, responsible for the sound generation and processing. In the *Handbook of Systematic Musicology* a following definition can be found:

> *The control space is concerned with the control variables input to any generative process, e.g., the oscillator frequencies, amplitudes, envelope shapes and modulation index in a frequency modulation (FM) synthesis model [...]* (Godøy 2018, 763)

In most circumstances, the control space is reflected in the actual interface design and determined by it. In the tools with a reasonable amount of parameters, most of these controls are low level and can be represented by physical handles on an interface, such as faders or sliders, or can be also accessed remotely via a messaging system (e.g. MIDI or OSC communication protocols). The one-to-one mapping of the input controls to the engine parameters is still the very common way of designing interfaces for musical production. But, as several researchers have proven (Hunt and Kirk 2000), although these kinds simple mappings may be practical for the sound design phase involving usually rational thinking, they are not so useful for a flexible real-time control over the desired sound results, or for the intuitive explorations of the timbre spaces.

> *One might then think that all the instrument builder needs to do is supply as many controls into the synthesis as possible. However, this can lead to a cognitive overload problem; an instrument may have so many controllable sonic parameters that performers cannot attend fully to all of them at once: they need a mental model simpler than brute-force awareness of every detail.* (Garnett and Goudeseune 1999)

Even if the dimensionality of the control and parameter spaces is equal, the researchers of electronic musical instruments design often mention that simple mappings of control values to synthesis parameters does not generate a natural feel or sufficient richness in the resulting sound. Therefore, other possible ways of mapping the parameters of the control space to the parameter space are usually tried, that go beyond the one-to-one mapping. In the more complex sound generation architectures we can find a number of strategies for overcoming the “cognitive overload” issue[25] arising from the plethora of available parameters. In these cases, the solution is to insert an additional mapping layer (or several of them) between the parameters and the physical interface. Tools such as ‘macros’ (weighted combinations of control connections) assignable to knobs, or

[25] See e.g. Mulder, Fels, and Mase 1997.

concepts of the ’modulation sources’ in a synthesizer with the possibility of diverse routings between them and the parameter ‘destination’ are common in the commercial devices and could be thought of as an extension of the *control space*.

One of the possible higher-level approaches of creating a meaningful relationship between synthesis parameters and the control interface would be to map the vectors in synthesis parameter space to the perceptual descriptors of the timbre space, and to control these by the movements in the timbre-informed control space. Using this idea, the Intuitive Sound Editing Environment (ISEE) project by Roel Vertegaal and Ernst Bonis assigned four higher-level timbre parameters to two X/Y surfaces, together forming a 4D *control space*: overtones, brightness, articulation and envelope.[26]

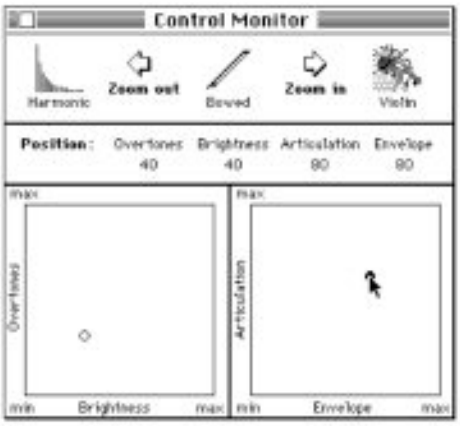

Fig. 11. The Intuitive Sound Editing Environment.

Of course, the idea of *control spaces* and their mappings to *parameter spaces* comes from a much broader domain of human-computer interaction, with many practical applications. For example, a system for an intuitive creation of facial animations could be based on drawing a trajectory in the control space (the colour circle), resulting in an animated human face illustrating transitions between expressions of different emotional states as seen in Figure 12 (Stoiber, Seguier, and Breton 2008).

[26] The authors actually call the ‘scaled implementation of the four parameters’ an *instrument space*. The reason for this is that the instrument controls also the loudness and the pitch of the respective instrument.

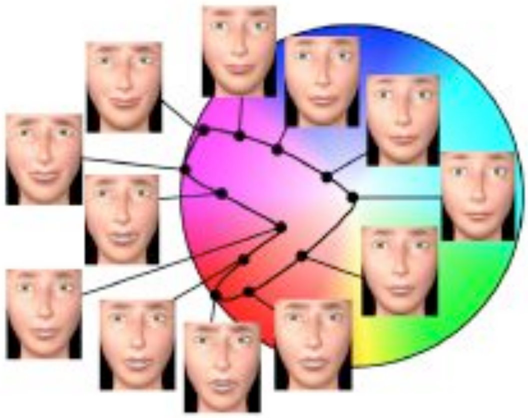

Fig. 12. Changing expressions of faces base on movements in the control space.

Is the *control space* identical with the *input space* defined by the input parameters determined by the possibilities of a user interface? Although in the simpler cases they may overlap, we can basically put an arbitrary number of layers (or, if we want, spaces) between the input data coming from the interface and the parameter space. A question of agency––i.e. who (or what) is in control?––also pops up here. Let us consider e.g. computational control spaces with modulation oscillators, particle system simulations, or software agents with some sort of sensing capabilities, all influencing the sound generation and reacting to the user input. Each of the cases is different, but it makes evident that the *control space* can become a quite complex concept, with the border between the *control space, input space* and *parameter space* not entirely clear.

3.6 Gesture, Input and Feature Space

A large part of the human communication relies on an arsenal of various kinds of expressive movements usually referred to as physical gestures. So, it is understandable that a whole branch of the HCI research focuses on investigating the various possibilities of a gestural control of computational processes. To achieve as intuitive and intimate control as possible, various sensor-equipped input devices have been designed to translate the user's gestures into streams of data for further processing. This is especially valid for the design of new interfaces for musical expression, since the gestural control has been traditionally synonymous with the notion of playing a musical instrument.

> *In stark contrast to the commonly accepted choice-based nature of many computer interfaces are the control interfaces for musical instruments and vehicles, where the human operator is totally in charge of the action. Many parameters are controlled simultaneously, and the human operator has an overall view of what the system is doing. Feedback is gained not by on-screen prompts, but by experiencing the moment-by-moment effect of each action with the whole body.* (Hunt and Kirk 2000, 232)

The types of physical gestures can range from static semiotic gestures of various kind to expressive temporal movements of a continuous nature which can have different length and structure (Gillian 2011, 79). In the musical context, since some gestures have a direct influence on the produced sound characteristics, others do not necessarily affect the sound and may have only accompanying expressive qualities. Such a variety of possible inputs has prompted the development of sophisticated computational methods of gesture recognition, including various machine learning techniques (Dynamic Time Warping, Gaussian Mixture Models, Hidden Markov Models, to name a few (Best, Bresson, and Schwarz 2018)) that are able to identify meaningful features in the input data in real-time.

A collection of all possible gestures, or, in other words, a set of continuous curves in a topological space, is being referred to as *gestural* or *gesture space*. According to Smalley,

> *Gestural space is the intimate space of individual performer and instrument. Performance gesture produces and defines a spatial zone within reachable space, the space being activated by the nature of causal gesture moving through that space in relation to the instrumental source, the whole event being united in the resulting spectromorphology.* (Smalley 2007, 41)

One of the ways of understanding the physical gesture spaces around a performer is to imagine a box around the moving body parts. Figure 13 shows how the isolation of the gesture spaces might be analysed. But it should be noted that to take into account the temporal nature of the gestures, we would need to add the time axis to the drawing.

Fig. 13. Various gesture spaces (Godøy 2010).

However, it is important to make one distinction here. In HCI terminology, the *gesture space* corresponds to the set of the data collected by an input device, instead of the set of possible movements in the physical space. Thus, based on the capabilities of the sensor, there can be sometimes a huge difference between the infinitely large number of possibilities of gestures in the physical space and gestures-as-data, filtered by the input device.[27] In some other cases though, e.g. with the use of a depth camera, the two notions can get closer.

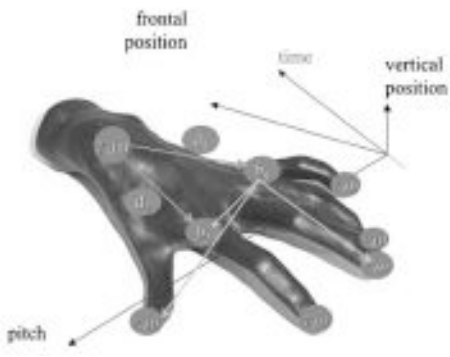

Fig. 14. The coordinates of a pianist's hand for its representation in a gesture space
(Mazzola and Andreatta 2007, 31).

[27] Compare e.g. a computer mouse with a 3D camera.

A spatial representation of gestures collected in time could e.g. look as illustrated in Figure 15:

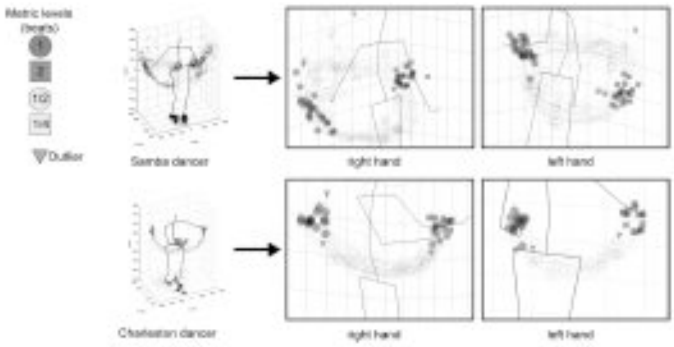

Fig. 15. Analyzed hand gestures of a dancer represented by a point cloud, corresponding to the music structure (Naveda and Leman 2010).

Besides the performer's gestures executed in the physical space translated to the gestures in the input space, it is interesting to imagine gestures that can be actually performed in any abstract topological space mapped to the sound generation. The composers can, for instance, design algorithmically generated trajectories in the *control space*. Related to this idea, another use of the term gesture refers to the sound or musical gesture manifested as movements of sonic structures in the perceptual sound space or, e.g. pitch space (Wishart 1996, 109).

> *Musical structures carry abstract perceptual features and characteristics that can be straightforward to identify by composers and/or listeners, but difficult or impossible to formally describe using the elements of standard score representations (e.g. identifying harmonic/melodic patterns etc.) Composers and authors often use the term of gesture to characterize these dynamic elements constituting musical forms, as an analogy with the idea of gesture in physical movements.* (Best, Bresson, and Schwarz 2018, 1)

One of the challenges in the design of the digital musical instruments is then to create a meaningful correspondence between the performer's gestures in the physical space to the resulting gestures in the *sound space* so that an illusion of their identity can be perceived.

Input Space

The term *input space* is used to denote an abstract space defined by the degrees of freedom of the input interface (or a combination of interfaces, in more complex cases). Although the interface can also capture gestures, because of their dynamic nature, the *gesture space* has an additional temporal dimension and, therefore, it cannot be usually considered a subset of the *input space.*

Feature Space

The movement in the *input space* can be directly mapped to the parameters of the sound generating process. However, with an increasing dimensionality of the *input space*, the mapping can become a computational or even conceptual problem. In such a case, a layer with various algorithms extracting higher-level features from the sensor data can be used after the input stream, as illustrated in Figure 16.

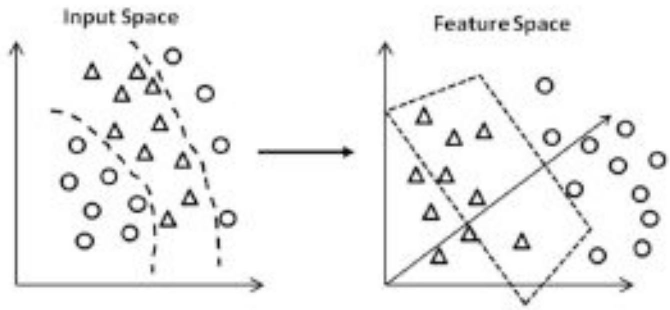

Fig. 16. Mapping of input data to a feature space (Pei et al. 2012).

The concept of *feature space* is often used in the music information retrieval (MRI) field including the machine listening and machine learning domain, where we can apply various dimensionality reduction techniques to optimise the training of the artificial neural network models. For example, to create a large but quickly searchable database of metadata extracted from many audio files, Tristan Jehan used MRI techniques to analyse every song in the database and reduce it to only a handful of features that still represent the key characteristics of the original (Jehan 2005).

3.7 Latent Space

In the recent years, the use of artificial neural networks (ANNs) in both the analysis and creation of music has been on the rise. The application field is wide, ranging from machine learning approaches applied in areas such as gesture recognition and complex input-output parameter mappings, to the so-called deep learning techniques, allowing for the generation of symbolic musical data (MIDI) or even high-quality audio (Oord et al. 2016). The ability of learning the characteristics of the data in the training set and use them to create new structures brings new solutions as well as issues, and it blurrs the boundaries between the machine and human creativity.

One challenge connected with the applications of neural network models in the creative domains––such as music––has been manifested in the task to generate new and coherent material, without actually repeating much of the structures contained in the training data but rather create meaningful variations (whatever that means). A recently discovered type of deep learning neural network structure called the variational autoencoder seems to achieve impressive results in this field.

Autoencoders are types of ANNs composed of several connected layers of neurons of different dimensionality, actually making up two symmetrical networks. The first network segment, called the Encoder (sometimes also called the recognition model), reduces the *n*-dimensional input information from the training data to a few-dimensional "hidden" middle layer. The other part, the Decoder (or generative model), then does the reverse: it tries to reconstruct the information to its original shape. This setup basically works as a compression/decompression algorithm. Because of the enormous dimensionality reduction, we will inevitably loose part of the information and some restored data will be not absolutely identical with the original. What is interesting though, the researchers found out that the low-dimensional middle layer, called the *latent space* can actually "learn" some high-level features from the input data. Based on this property it turns out, that when we present new input data to the network (i.e. some that were not contained in the training set), the model can still output a meaningfully looking––or sounding––prediction.

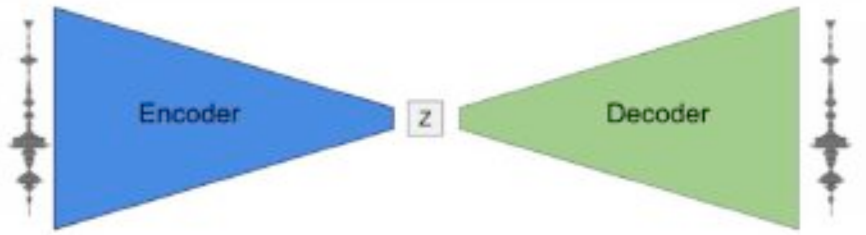

Fig. 17. The structure of an autoencoder (Roberts, Engel, and Eck 2017).

In case of the special network subtype, the variational autoencoder, we can do different vector operations on the vectors of the *latent space*, which, after decoding, technically result in variations of the learnt data representations. Because of the low dimensionality and high information density of the *latent space*, it is easy to achieve complex changes of the output by operations (e.g. movement) in this layer. The big advantage is that in this way, the output data still maintain some high-level coherence.

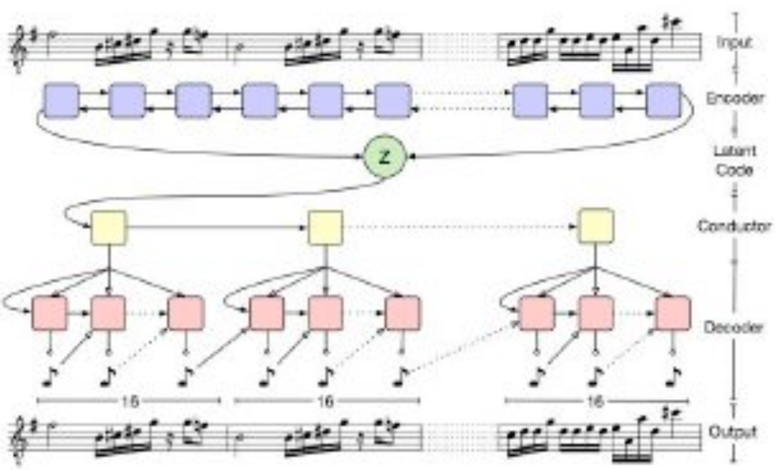

Fig. 18. Schematic of the MusicVAE autoencoder model. In this case, there is no difference between the training data and the prediction (Roberts, Engel, Raffel et al. 2018).

Based on the variational autoencoder types of neural networks such as the Google Magenta's MusicVAE, trained on large datasets of symbolically encoded musical information, we can build applications with easy-to-use interfaces able to produce variations of drum patterns or melodies in real-time (or almost real-time) situations (Roberts, Engel, Oore et al. 2018). For example, by generating a series of interpolations between two points sampled from the latent space, it is possible to create gradual transitions between different musical motives, or even between polyphonic music passages of different genres. In this manner, the process of composition can become the design of trajectories between the points in the *latent space,* resulting in a musically meaningful "evolution" of the original material. Figure 19 is an excellent example of showing how the latent space can be used in developing complex yet intuitive gestural interfaces, whre the changes of a latent space vector gets reflected in the output drum pattern.

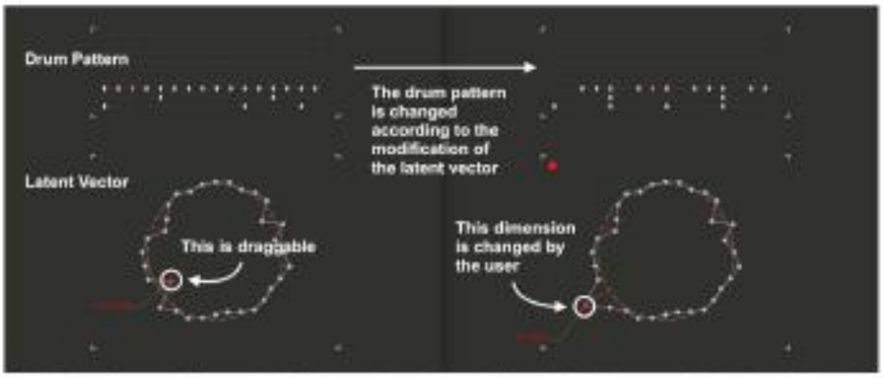

Fig. 19. The Latent Inspector (Thio et al. 2019).

3.8 Conceptual Space

In brief, my instrumentalist standing means that I eschew philosophical discussions of how "real" conceptual spaces are. The important thing is that we can do things with them. (Gärdenfors 2000)

The thoughts about the various ideas of spaces bring us to the following questions: what do all the aforementioned different concepts of space have in common and where does the geometrical nature of cognitive representations originate, at all? When researching this problem across multiple domains, the cognitive scientist Peter Gärdenfors came to several valuable conclusions. He found out that the spatial representations with various degrees of abstraction are deeply rooted in the way the fundamental human cognitive processes work. According to his findings, much of the information in the human brain is represented in a geometrical, rather than a symbolic way and many models of concept formation and learning are based on spatial structures (Gärdenfors 2000). In accordance with our knowledge about spatial ways of thinking in music, he also suggests that there are certain types of cognitive problems for that using topological representations on the conceptual level appears to be efficient. Besides the symbolic and connectionist models of cognition that had been dominating the discourse, he introduced the notion of a *conceptual space*. Such spaces are cognitive structures for organising information, comprising one or more quality dimensions and sorted into domains. There are basic domains with qualities linked to human sensory perception (such as colour, sound loudness, or the three dimensions of a perceived physical space), but they can be also more abstract.

The efficiency of geometrical models has been proven for both the analytical and constructive approaches to cognition, with the border between them not being sharp. We can see this overlap also in the design of digital instruments and creative systems for music: for instance, the visual spatial representation of sound grains in corpus-based concatenative synthesis is based on the analysis of the sounds and so it informs the performer's predictions and gestures. Gärdenfors calls this a spiraling interaction between the explanatory and constructive uses of the *conceptual space*, which are the two main axes the cognitive science research evolves around.

In his theory, he furthermore distinguished between three levels of representation with variable dimensionality or "resolution": the subconceptual, the conceptual, and the symbolic. It becomes evident that we could frame some of the notions of space in music examined in the previous sections as belonging to either of these categories., which suggests that geometrical structures are functional on all levels of representation.[28] We could consider many of the perceptual spaces rooted in the auditory scene analysis, such as Smalley's notion of the perspectival space, subconceptual, since they

[28] It is important to note that *conceptual spaces* are continuous and low-dimensional, which might not be true for the spatial representations on the other two levels.

result from a low-level cognitive processing. *Timbre space* would then be viewed as an upgrade to a conceptual category, whilst *parameter space*, *control space, feature space* and *latent space* would be abstractions on the symbolic level where e.g. computational operations could be used.

> *The representations on the three levels occur at different scales of resolution: a high-dimensional vector on the subconceptual level is reduced to a low-dimensional structured vector on the conceptual level; and a symbol just summarizes the information contained in a region of a domain of a conceptual space by referring to the prototypical element of the region.* (Gärdenfors 2000)

The research suggested that we can derive the geometric structure within one domain from information gathered on the subconceptual level. If we apply this knowledge about the human perception in the HCI field, it is possible to do this with the help of various computational processes such as the self-organising maps or other types of feature extraction algorithms. As mentioned before, in this way we can e.g. create comprehensible, real-time updatable low-dimensional spatial map of distinct timbres based on the machine-based analysis of the sound characteristics.

The identification and separation of the various domains can be useful in creating relations between them, such as mappings between geometrical structures. In relation to this, some note that "*mapping structure from a nonmusical domain onto music is a way of creating musical structure, and different mappings will lead to different accounts of musical structure.*" (Zbikowski 2002, 14).

These principles could be generalised for the connections between elements of any kind of spaces, including, in our case, technological abstractions used for the creation of new musical instruments or interactive systems, where the three distinct levels can also take up different tasks (Gärdenfors 2000).

As it has been shown recently, the application field of the *conceptual space* in music is suitable for explanations or constructions of different kinds of creative systems, including, e.g. the musical practice of live coding (creating music by writing and executing computer commands in real-time). Figure 20 is a symbolic illustration the use of Tidal live coding language as a tool for searching in the *conceptual space* of music (Wiggins and Forth 2018):

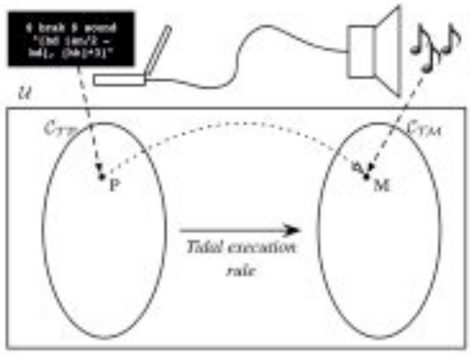

Fig. 20. Live coding process in the Tidal environment (Wiggins and Forth 2018).

A big advantage of spatial representations is that allow us to think in terms of proximity, centroids, regions and other geometrical concepts when actually operating with much higher-dimensional information. However, it is still necessary to distinguish between the psychological and scientific applications of the concept, depending on the nature of the spaces in question. We can find a similar notion of *conceptual space* in the seminal book of Margaret Boden *The Creative Mind*. In her general model of creativity the space is explored by creative agents.

> *The identification of conceptual spaces isn't an exact science. To be sure, it has to be made exact if the spaces are to be reproduced in a computer program … But conceptual spaces in real minds aren't always so cut- and-dried. One could say that they are idealizations. However, like the 'ideal gas' in physics, they are very useful to people (psychologists, not physicists) trying to work out what is going on.* (Boden 2003, 74)

As an example of the constructive side of the *conceptual spaces*, Jamie Forth, Alex McLean and Geraint Wiggins designed a creative system for music making based on the concept of Wiggins' Creative Systems Framework extending the Boden's idea of agents' spatial movements. The proposed model as seen in the Figure 21 uses the movement in timbral and rhytmic conceptual spaces corresponding to the generation of musical structures in the perceptual spaces––in this case, variable rhythms of percussive timbres (Forth, Mclean, and Wiggins 2008).

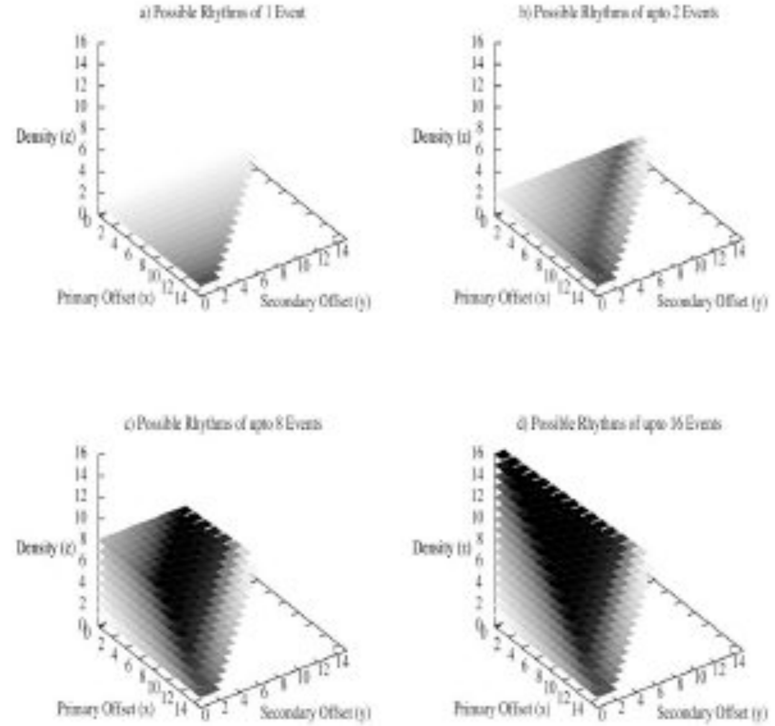

Fig. 21. Rhythm space (Forth, Mclean, and Wiggins 2008).

The *conceptual spaces* can be used as a tool in the design of new musical systems. But also, as Boden sums up, they play an important in the whole creative process:

> *In general, the more types of concept and conceptual space that can be built, and the more flexibly and fruitfully they can be combined, explored and transformed, the greater the understanding—and the greater the creativity. Computational psychology has provided a host of theoretical ideas with which to consider novel combinations of concepts, and with which to map the conceptual spaces constructed within human minds.* (Boden 2003, 294)

3.9 Shifting Possibility Spaces

In his application of game theory to the musical domain David Kanaga draws an interesting parallel between a virtual game space with its possibilities and a space of musical creation. As the author notes, the medium of a game,

> *"has the capacity to encompass and integrate all playable forms—all interactive algorithms—which computers are able to embody. This is a totalizing effect which requires different metaphors. The best may be the image of games as opera, following George Lewis's theorization of interactive computer music, writing that 'interactivity suggests a new model for the Gesamtkunstwerk, one which is wary of hubris and disinclined to overweening centralization strategies' (Lewis 2009, 460)."* (Kanaga 2018)

Kanaga introduces the idea that games and music are both "playspaces" based on the same principle. While this could be applied to the music making processes (including composition or improvisation) in general, we could think of the interactive music systems as being even closer to the video games where the whole game is regarded as a piece of music, or instrument, or both.

Even in terms of the working environments, the border between the creation of an interactive system for an installation, a musical performance or a computer game is fuzzy. Many present interactive artworks are created in game development software,[29] where the typical soundtracks have the form of––in the game terminology––procedurally generated audio. Computer games are interactive systems with intelligent software agents, which is another shared feature with creative musical systems. Undeniably, one of the common grounds here is the nonlinearity and the open structure of the experience. In one of the games designed by Kanaga himself, there is a quite literal illustration of the "playspace" principle. In one scene the player can play a piano score by drawing on it: the notes being "touched" by the virtual pencil actually start to make sounds.

A related concept of a *possibility space* originating in mathematical statistics and physics (sometimes also called solution space or probability space) has become a familiar term in game theory and development, referring to the sum of possible moves of interactions in the current game space state.[30] From an interactive systems design point of view it is interesting that Ian Bogost uses the

[29] Such as Unity (https://unity.com) or Unreal (https://www.unrealengine.com).
[30] This can, of course, relate to any type game, not exclusively video games.

notion of gesture when claiming that *"the possibility space of play includes all the gestures made possible by a set of rules."* (Bogost 2008). Even more interestingly, he continues by a generalisation of the idea to the domain of artistic creativity: *"In more traditional media like poetry, the possibility space refers to the expressive opportunities afforded by rules of composition, form, or genre."* (Ibid., 121). This definition can be, of course, applied to the creative musical systems as well: playing the system, just like playing a game, means exploring the numerous configurations. When studying the *possibility spaces* from a philosophical point of view in different contexts, Manuel DeLanda adds another valid point about their possibly variable nature: *"some possibility spaces are continuous having a well-defined spatial structure that can be investigated mathematically, while others are discrete, possessing no inherent spatial order but being nevertheless capable of being studied through the imposition of a certain arrangement"* (De Landa 2011).

If we add a temporal domain to the system or if we consider the possibility of changing the external conditions or its inner structure, we could further elaborate on this idea by introducing the formal tool of *shifting possibility spaces* as suggested by Kanaga. This is yet another approach to describe or design time-based experiences through a spatial metaphor, where the number of possibilities changes with the situation:

> *Game designers often speak in this way about the totalizing 'possibility space' of a game, in the same way a music theorist might speak of a piece's form (e.g. sonata-allegro, fugue), but what is lost in this global analysis, especially in the case of musical games, is an acknowledgement of the temporal flux of shifting possibilities, based on the contingent value of what is possible for a player at a given moment. Playing is a process of moving through possibility spaces. Considered locally, the experience of a possibility space is not that of a solid object but rather of a morphing form, with shifting presences and absences of free variables corresponding to shifts of local dimensionality.* (Kanaga 2018)

The dynamic nature of the *possibility space* during the process is an important feature when thinking about music improvisation. If we move on from the descriptive level to the construction of generative algorithms for a real-time musical performance, the principle of *shifting possibility spaces* could be also applied even in a more direct sense. For instance, instead of a fixed mapping between the various computational spaces (e.g. *feature space* -> *control space* -> *parameter space*) during a performance, we could achieve a greater flexibility, i.e. expansion of the "generative space" or "phase space" representing all possible states of a system, if we attempt to design a changing environment in which the same movement in the gesture space would yield different sonic results.

A very basic and often used approach would be to design different ”modes” for the input device that the performer can choose between: an option to switch the context is common in many commercial MIDI controllers. But we can also think about more advanced methods, like the gradually evolving topography of mapping layers with changeable weights for the parameters, responding to the position on a timeline, or on an external input. The idea of the enhancement of parameter mapping strategies beyond the static layout has been explored by several researchers. For instance, Ali Momeni and Cyrille Henry achieved such a time-varying behaviour of mapping algorithms by inserting an algorithmic layer containing a mass spring physical model or creating dynamic interpolation spaces (Momeni and Henry 2006). Another example of a flexible mapping technique could be a dynamic modulation matrix for the sound synthesis parameters as used in the Hadron Particle Synthesizer (Brandtsegg, Saue, and Johansen 2011).

In this chapter I have investigated several concepts of spatial thinking used in the theory and practice of electronic music, relevant for the construction of creative musical systems: *music space, sound or sonic space, timbre space, parameter space, control space, gesture space, input space, feature space, latent space, conceptual space,* and *shifting spaces of possibility*.[31]

The diagram in Figure 22 suggests possible relations of these spaces and how they can be connected in a model of an interactive music system, including the options of inputs from both human and non-human agents.

[31] For practical reasons I did not discuss some of the other possibly relevant terms, such as sound-oriented virtual reality spaces or the phase space, describing all possible states of a system. I also omitted some spaces that are not topological vector spaces with uniform characteristics.

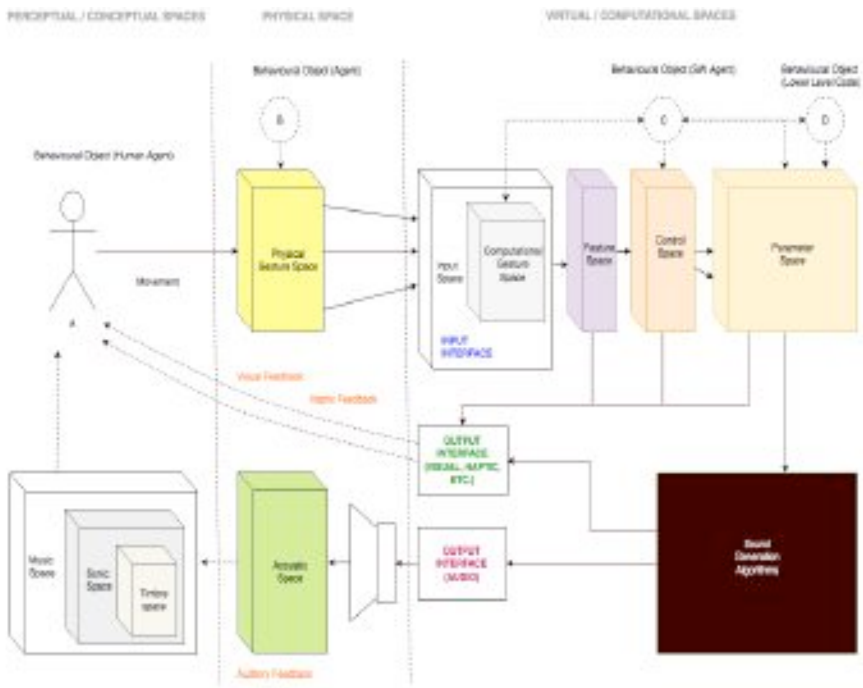

Fig. 22. Various types of spaces and connections in an interactive music system.

In the drawing we can see a rough division into at least three categories of spaces, based on their ontological nature: the physical space, perceptual and conceptual spaces, and computational spaces. In reality, the borders are not always as clear, the spaces can be nested or overlapping and some layers of the information processing chain can be missing or optional. These spaces can have a different shape, and so the connections between them often require some dimensionality reduction and complex mappings between variables. The mappings can be, of course, dynamic, i.e. changing over time, further expanding the possibilities of the system. This is a basic schematic, but, of course, an arbitrary number of computational layers may be added to the system.

In general, we can identify several levels on which the actors, called also agents or behavioural objects, can interact with other parts of the system, or with each other. As noted before, these agents can be of various kind, e.g. physical objects (anorganic, organic, technological, etc., marked as B), moving in the physical space and being captured by the sensors, one or more human performers (A), but also software agents of different abilities (C) or other software code we could also attribute some sort of agency (D). The sound generation algorithm is interchangeable, depicted as a black box equipped with input and output interfaces.

Based on these ideas, building digital musical instruments, interfaces and performance systems could be understood as designing and linking spaces of various dimensions and ontologies, containing objects with different agency, and thinking about the movements and interactions in these spaces. In the next chapter I will therefore various strategies of navigation in such spaces and explore their possibilities especially in respect to the real-time music generation in a performance context.

4 Performing by Navigating

Movement makes time emerge from the experience of space. (Mylov 2002, 47)

In the previous sections I have already mentioned some of the issues appearing in the literature related to the design of digital musical instruments and interactive systems (Pendharkar, Gurevich, and Wyse 2013). Although the border between these two concepts is sometimes not very clear with most of the tools lying somewhere on the notorious ”Instrument vs. Player” continuum, some of the recurring topics might be generally relevant:

1. Simple parameter mappings between the input gestures and sound output do not work well and usually feel unnatural even for the design of simple digital instruments, not mentioning more complex interactive systems. (Hunt and Kirk 2000)

2. Inability to meaningfully work with a large number of control parameters (Ryan 2002)

Multidimensional parameter spaces of sound synthesis can be huge and therefore hard or impossible to search intuitively. Most of the ”locations” yield not necessarily musically very useful or interesting sounds. On the other hand, constraining the spaces by chosing a less complex synthesis algorithm leads to a paradoxical situation as observed by Palle Dahlstedt:

If the sound space is to small, the listener will know it all after a short period of exploration and loose interest in the work. On the other hand, it will be difficult to navigate if it is too big. The paradox is that the more universal the algorithm, the bigger the space of possible results, and the lesser the good-sounding fraction of it. (Dahlstedt 2001, 3)

4. If a sonically more complex evolutional process is desired, a further extension of the computational space is needed to take care of the temporal properties of the process (in technical terms e.g. changing modulations or timing of events).

These findings suggest that a sophisticated control structure is necessary for generating dynamic sound morphologies. Yet even if we stick to the synthesis domain alone and leave up the "movement" part to the performer––which is usually the case with simpler instruments––, a fast and intuitive access to the points in the parameter space remains a key design task.

4.1 Presets

> *There is a cultural consensus that creativity is applied to a specific set of parameters, while others are taken for granted. Presets of the mind.* (Goldmann 2015)

A typical method in which different electronically produced timbres can be made instantly available is to find or handcraft a certain configuration of parameter values––a point in the parameter space with its perceptual equivalent––and save the coordinates as a preset. With the development of a patch memory capable of storing such data introduced in the early 1970s, this method has been widely established since the 1980s as an industry standard in the synthesizer design in both the hardware and software worlds. Presets are "digital shortcuts" enabling the musician to get an instant access to the desired "sound" (Goldmann 2015). This technology has contributed to the emergence of some iconic sounds with specific timbral qualities, widely used most of all in various popular music genres. The concept of "factory presets" included with the product by default and demonstrating the device's capabilities and sound-designer's skills has also opened a shared access to the music making with electronics to a wide user-base. Presets have become a cultural phenomenon with a memetic agency on their own: often they are so useful and fitting into a particular musical style that they even get never modified by musicians.

However, it is obvious that the concept of presets as used in the commercially available synthesizers continues to follow the quantisation paradigm of the acoustic instrumental world. It is undeniable that even aside of the industry-driven instrument design the idea of presets has become an inevitable way of quickly accessing particular sound qualities. Yet for the design of a music performance system the possibility of storing and recalling presets is usually only the beginning of a journey, not the end. Presets can store and recall a desired configuration but they do not solve other problems. What if we, for instance, want to access timbres that are located "between" the two––or more––given sonic identities? One of the interesting possibilities arising from the nature of the geometrical representations in the conceptual space is to think about the “betweenness” as an attribute of the structures:

> *The geometrical nature of the conceptual layer means that it is literally possible to ask questions like “what concept is one third of the way between these two others?” and to get a range of answers that are meaningful.* (Wiggins 2018b, 14)

The concept of searching between the known configurations can be also applied in the exploration of sound synthesis parameter spaces. For instance, some of the interesting non-real-time approaches for timbral exploration include the use of ”genetic algorithms” for preset evolution, based on the evolutionary principles of ”mutations” (meaning random changs in the settings) and inheritance of synthesis parameters by means of interpolation between the two parameter values of selected “parents” (Dahlstedt 2001). In this method, the selection and evaluation of the presets generated in each generation are done by a human listener.

Coming back to the related ideas of timbral continuum and sonic fluidity developed earlier, I will continue with the approaches allowing for a deeper exploration of the available sound spaces in a real-time context.

4.2 Metaphors of Navigation

The idea of navigation in various kinds of spaces is popping up frequently in the electronic music discourse and specifically in the research related to the design of creative music systems. In such cases, the gestures of the performers create trajectories facilitating the exploration of different timbral possibilities, melodic or rhythmic structures:

> *One of our central metaphors for musical control is that of driving or* ***flying about in a space of musical processes****.* (Wessel and Wright 2002, in Collins 2006, 13)

> *Seeing a computation as a path in some abstract space is hardly new: the representation of the execution of a concurrent program as a trajectory in the Cartesian product of the sequential processes dates back to the 60’s [...] However, [...] the considered space is based on the control structure, not on the involved data structure.* (Giavitto and Michel 2002, 11)

Yet as the research overview about conceptual spaces has already demonstrated, there is also a more general concept of searching in the spaces of solutions that is used in the creative processes but also e.g. in engineering terminology. The similarity between the search in a conceptual or solution space and the navigation through the sound space is therefore not coincidental. Coming back to the common cognitive basis of "movements" in ontologically different spaces it is evident that navigation seems not only as a common and appropriate metaphor to describe these processes but also a useful tool helping to actually create a design concept for musical use. However, it is especially worth mentioning that in contrast to the engineering-type of searches, in improvisatory musical practice the navigation can be guided by intuition and exploratory needs thriving for new sonic constellations, as being part of the divergent phase of a creative process.

4.3 Interpolation Spaces

In order to be able to navigate the sound space in an intuitive, expressive and exploratory way, a favourite option is to create a so-called few-to-many or many-to-many mapping connection between a low-dimensional input interface allowing for gestural input to a usually higher-dimensional parameter space of the sound synthesis engine. The resulting sonic gesture is then a correlate of the movements in the respective spaces. Since gestural movements in a physical space with two or three dimensions are natural to humans––albeit more-dimensional interfaces exist as well––this is an obvious method of choice. Once the input data form an interface get captured, the desired effect of dimensionality scaling between different computational spaces can be achieved by dimensionality reduction techniques, projecting a point in the *input data space* to a point in the synthesis *parameter space.* Various interpolation techniques have been tried out in the electronic music throughout the past decades (e.g. inverse distance weighting, natural neighbour, or intersecting N-spheres, see (O'Sullivan 2013), with its applications ranging from basic to complex[32]. Some of the most interesting results can be obtained by the use of interfaces that can do interpolation between stored parameter configurations:

> *Hybridization interfaces are powerful tools for managing the large and ever growing number of control variables available to the artist and performer working on image, motion, and sound*

[32] Already in 1992 Lee and Wessel *"have successfully trained a neural network to generate parameters for several synthesis models with timbre space coordinates as input, automatically providing timbral interpolation."* (Vertegaal and Bonis 1994).

synthesis. Using these interfaces, we navigate through a low-dimensional control space generating weights to combine multidimensional parameter sets from a small number of interesting data points. (Freed et al. 2010, 343)

The notion of "hybridization" is a bit confusing here though, since it is used as *"a way to place emphasis on the new forms that arise as the user transitions from one data point to another, rather than on the data points themselves" (Freed et al. 2010, 343)*. However, the essential idea is that "*once a set of presets has been created, a lower-dimensional sub-space can be created from them, the simplest being a line that interpolates between two preset points" (R. Tubb and Dixon 2014, 24)*. *Interpolation spaces* of this kind have "condensed" gravity points witch expand to points in multidimensional *parameter space*. Due to the nature of this dimensionality reduction method, only the preseleted and "most interesting" parts of the prameter space are made accessible.

4.4 Interpolators as Control Structures

In terms of storing, recalling and working with the predefined parameter configurations, several inspiring tools were developed based on the arrangement of points or areas on a plane representing stored preset data. An early example of the multidimensional sound space continuum controlled by a 2D interface is the ***Syter*** instrument (shown in the Figure 23) developed at INA GRM in the late 1970s and early 1980s (Teruggi 2007). A suite of audio effect plugins GRM Tools developed in the 1990s has also a built-in interpolation option, but it is only possible to morph between the settings on one axis.[33]

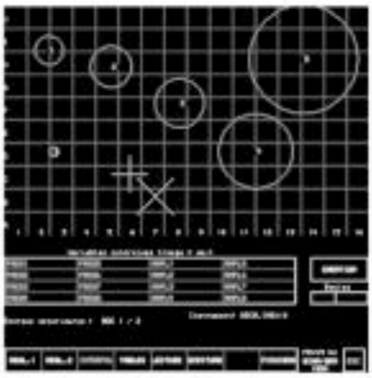

Fig. 23. The interpolation screen in Syter (Teruggi 2007, 225).

[33] https://inagrm.com/

Since then the design of many control tools in different programming and sound production environments has been inspired by the two-to-many mapping idea, as well as some complete software instruments integrating user control with the actual sound generation. Among the former is the Ross Bencina's ***Metasurface*** tool, a part of the *Audiomulch* audio software (Bencina 2005). It uses a similar approach in a visually appealing design: it enables to create arbitrary zones with user-stored parameter configurations and the ability to morph between the values by the movement of a mouse cursor (Figure 24). The points have to be however created beforehand, outside of the performance time, because of computationally demanding calculations of the "nearest neighbours" for interpolation. The Metasurface was anyways an important inspiration for my design thinking since it enables to store configurations of different kinds of parameters, not only for sound synthesis. It's use is, however, limited to the Audiomulch environment.

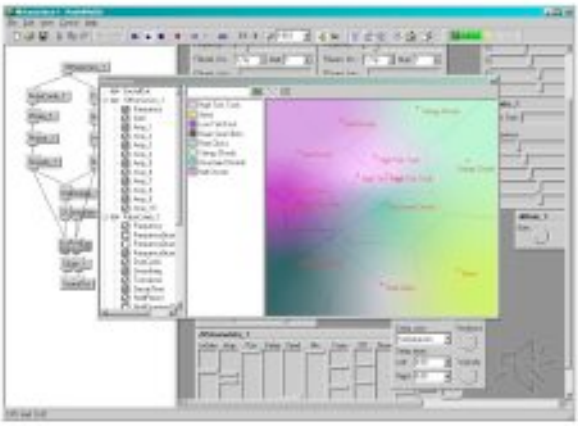

Fig. 24. The Metasurface component in Audiomulch software (Bencina 2005).

Ali Momeni and David Wessel (Momeni and Wessel 2003) present an overview of even more historical examples of the plane navigation approach to the simultaneous control of multiple parameters. Following on their research they conceived an interpolation tool called ***space-master***, created as a patch within Max/MSP environment shown in the Figure 25, using a technique that they call mixture-embedding:

> *Instead of preserving the geometry of an input space, a mixture embedding creates the structure by user-defined associations between input and output states. In doing this, there is an implicit*

perceptual distance imposed on the two spaces: Something that is considered by the designer as "close" and grouped accordingly may actually be quite far in the Euclidean space of sound parameters. (Momeni and Wessel 2003)

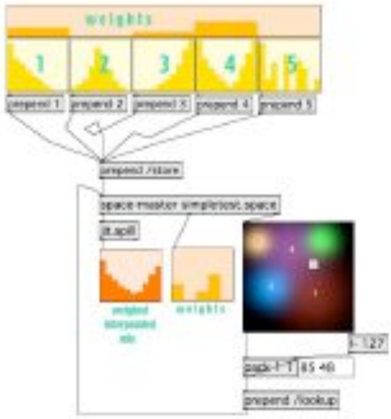

Fig. 25. The *space-master* Max patch interpolating between the settings of 5 lists by a movement I in 2D control space (Momeni and Wessel 2003).

A similarly intuitive graphical interface is Oli Larkin's ***pMix***[34] ("preset mixer") which makes it possible to interpolate between the parameters of several software synth plugins simultaneously (VST2, VST3, AU or LADSPA, FAUST script). The project originally started as ***int.lib*** (interpolation library) for Max/MSP environment. This is an approach of a polyphonic design, where the parameters of multiple plugins can be controlled simultaneously.

The ***Influx***[35] class for thes *SuperCollider* environment developed by Alberto de Campo offers a yet different approach to the dimensionality reduction problem in the interactive instruments. It creates a matrix of connections where

*any number of named control parameters can be mapped onto any number of named process parameters, by having a matrix of weights for the amount of influence of each control parameter on each process parameter. (*de Campo 2014)

[34] https://www.olilarkin.co.uk/index.php?p=pmix
[35] https://github.com/supercollider-quarks/Influx

For instance, in the example in Fig. 26, cursor movement in a 2D control space sends control data used to influence the 16 desired parameters based on a number of adjustable weights.

Fig. 26. GUI with 2D slider, Inlux, KtlLoop and Preset controls. From (de Campo 2014).

The weights can be set either arbitrarily by hand or at random with some additional options which often leads to interesting relationships between objects and, in turn, also to surprising sonic results:

> *with every new set of weights, a different subspace of the overall state becomes accessible, allowing one to find different sweet areas; more entangled mappings create more complex changes even with simple movements, which may be more interesting to play [...]* (de Campo 2014)

This approach is really unique especially in that it allows for creative exploration of parametric subspaces driven by randomness.[36] Moreover, the *Influx* can be, of course, used in connection with with other classes (e.g. recording and playing back input gestures) to build complex interactive architectures of networked influences potentially involving several agents, including multiple human performers as depicted in Figure 27.

[36] I will get back to de Campo's concept of Metacontrol in the next chapter.

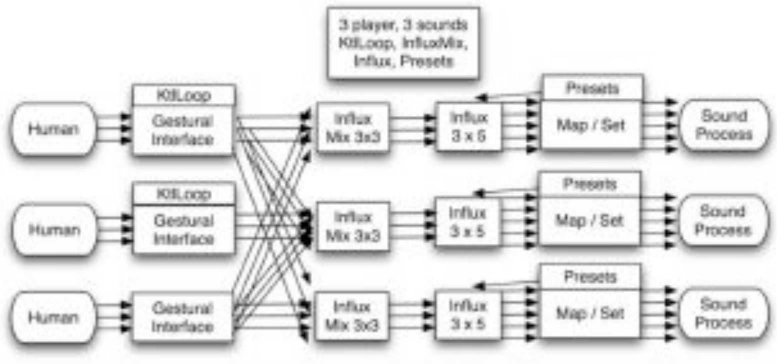

Fig. 27. Possible performance system architecture using *Influx* (de Campo 2014).

4.5 Machine Learning

Since quite recently, machine learning techniques have been used as interactive dimensionality reduction mapping techniques capable of learning the trajectories between preset points in high-dimensional spaces on the fly. There are several available free and open source software tools that are very flexible and can be easily used by the artists, musicians and instrument builders in the "mapping by demonstration" manner. This can be illustrated by the case of supervised learning algorithms in the *Wekinator* program by Rebecca Fiebrink (Fiebrink 2017) or other examples such as *ml.lib* machine learning library for Max and PureData, programmed by Jamie Bullock and Ali Momeni (Bullock and Momeni 2015).

4.6 Synthesis Interfaces

In the world of commercial applications, Wolfgang Palm introduced an innovative "Sound Map" concept applied in his software synthesizers ***WaveMapper*** and ***MiniMapper***.[37] Here, the user can move several visual objects different locations and interpolate between various settings available separately for individual modules of the sound synthesis engine (oscillators, filters, etc): "*The Mapping window is the visualisation of a pool of programs on which 8 map icons float. Those icons represent certain parameters of the synthesizer engine and each icon can be placed on one of the 32 map programs. From that moment on the parameters of that icon will take on the settings of those*

[37] http://wolfgangpalm.com/iwm.html

parameters in the underlaying map program in the MAPPING module." ('Wave Mapper Manual.Pdf' n.d.)

Fig. 28. MiniMapper app for iOS.

4.7 Sequence Morphing

In most cases, the control surfaces with topographic representation of the stored values are used for expressive modulation of synthesis parameters. However, the principle of interpolation can be also extrapolated to other parameters of a performance setup. In addition to that, we could also imagine morphing between sound structures besides merely modulating the timbral properties of the sound. This can be done with operations on the symbolic level, such as MIDI or other type of data, before the actual sound generation. Daniel V. Oppenheim described this process as compositional morphing:

> *Audio mixing can be regarded as a limited case of morphing involving only one musical element: volume. Compositional morphing will typically transform several other elements, namely pitch and rhythm. Another fundamental difference is that all intermediate stages of a mix include all the musical elements from all the mix inputs, whereas a morph will produce a single sequence of musical elements that is derived from all music inputs.* (Oppenheim 1995)

Oppenheim's software ***DMorph*** was designed with the aim to provide such functionality, however, the timbral morphing was done through mere crossfading between the sounds.

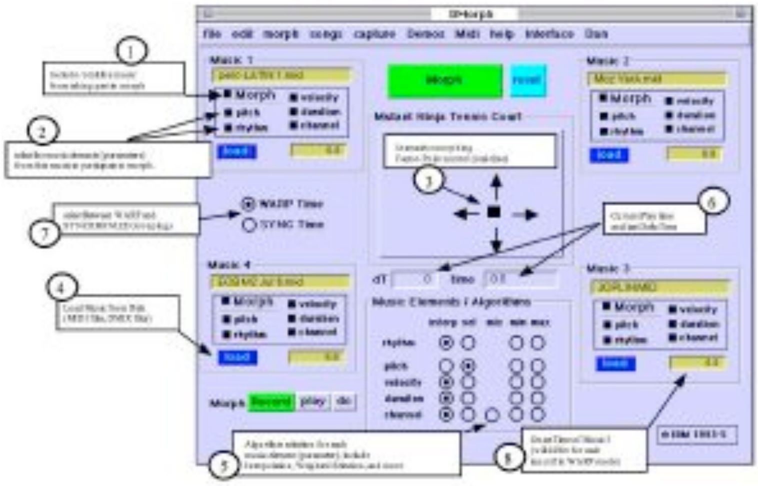

Fig. 29. DMorph graphical interface (Oppenheim 1995).

There have been several approaches presenting interesting solutions to the problem of smooth structural morphing between two or more time-based sequences, such as rhytmic sequences, melodies or the whole songs. Whilst not all of them offer the possibility of a real-time interaction, some of the more recent ones are relevant for the concept of a topographic interface. For instance, René Wooller and Andrew R. Brown (Wooller and Brown 2005) discuss some further applications for morphing understood as transition between two different musical sections (represented in symbolic data), including some historical examples that are not available as functional software anymore. The authors make a distinction between different kinds of morphing strategies for time-domain transitions useful for procedural music generation e.g. in computer games.

The app ***Gestrument***[38] for iOS devices conceived by composer Jesper Nordin offers intuitive gestural control over (quantised) polyphonic rhytmic and melodic sequences with adjustable proportion of probability and randomness. Apart from its own synthesizer engine the instrument also outputs MIDI data that can be used to control other software or hardware synthesizers.

[38] https://gestrument.com/

Fig. 30. Gestrument iOS app.

4.7 Hardware Modules

An interesting example of navigation in the virtual space coming from the world of modular synthesizers is the *Grids* module in Eurorack format by Mutable Instruments, marked by its creator as ”topographic drum sequencer”. It generates patterns of control voltage triggers based on the imaginary position on a map, where “*thousands of variations can be intuitively generated by controlling the ’event density’ of each of the 3 channels (bd, sd, hh)–-gradually moving from a sparse backbone to a deliciously rich pattern with ghost notes, rolls and fills.*”[39] The sequence morphing technique embedded in the module is based on a principle of a map connecting points in a two-dimensional space to a higher dimensional rhytmical space with pre-programmed patterns.

[39] https://mutable-instruments.net/modules/grids/manual/

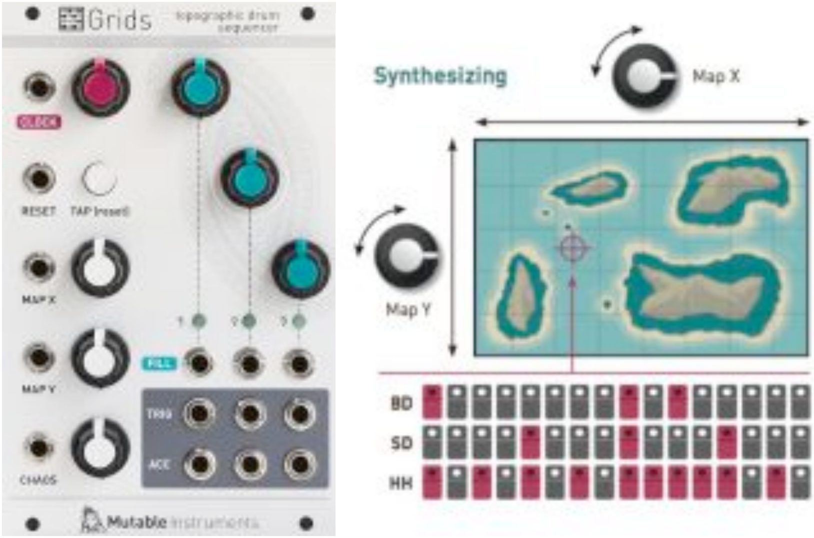

Fig. 31. The Grids control voltage module and the illustration of the principle of a "map" containing rhytmical patterns. (Grids Manual)

Another Eurorack utility module is the ***Vector Space*** module developedby Worng Electronics,[40] using a geometrical concept of a virtual cube to distribute a control voltage signal to 17 different outputs based on a three-dimensional input.

Fig. 32. The Vector Space module.

[40] https://www.worngelectronics.com/

Over the last three decades, several interpolation tools have brought innovative solutions for the navigation and exploration of complex sound spaces. All the aforementioned design strategies illustrate how the idea of gestural movement through user-defined topographies can contribute to the overall fluidity of the resulting sound process. For practical reasons I did not mention yet another complex and interesting subject: the design of interactive tools for sound manipulation in virtual visual environments, where the border between a musical instrument and sound-oriented video game is becoming really fuzzy. This very appealing field of research that would be worth a dedicated exploration. In a continuation of the search for a greater amount of real-time controllable complexity, many ideas from these concepts could be further developed to facilitate the generation of dynamic morphologies in real-time, which I will demonstrate in the next chapter.

5 AMEN – The AMbiguity ENgine

5.1 Topographies of Behaviours

Alberto de Campo's concept of Metacontrol, summarised with the claim *"Lose Control, Gain Influence"* proposes to delegate the full control over low-level processes to the algorithms in favour of a high-order influence on their behaviour (de Campo 2014). He quotes the authority of Joel Chadabe, who states that

> *[t]he primary benefit of an electronic instrument for a professional performer, which is that it extends the performer's capabilities in interesting, creative, and complex ways, requires an intermediary mechanism between gestural control and sound variable* (de Campo 2014)

In my proposal of a music performace system I am extending this idea by suggesting a design based on topographies of sound generating algorithmic behaviours influenced by a navigation in a three-dimensional *interpolation control space,* with additional control data inputs from various sources. The main goal was to create a cognitively accessible control space, a computational environment for the creation of evolving sound morphologies in real-time. The system offers the possibility not only to get to the various points in the high-dimensional parameter space in a quick and intuitive way, but also to continuously morph between various ever-changing sonic identities. The improvisatory process can then be developed from there on, into the "spaces of possibility" imagined "orthogonal" to the location in the original space in a higher-dimensional manifold. This could be done in several ways: by navigating between the dyanamic algorithmic behaviours and simultaneusly influencing the high-level parameters of the event generation via gestural input, making decisions about the temporal operations with the material (such as repetition of phrases, automatic evolution of "freezing" of the motion at a desired sonic state, etc.). An important feature is the option to vary the amount of automation and autonomy of the system's response, which leads to the exploration of interaction possibilities in different scenarios, ranging from solo and "AI-assisted" performances to collective improvisations.

5.2 Design Considerations and Development Process

The concept of navigation in high-dimensional paramter spaces introduced in the previous chapter has multiple benefits for improvisatory performance situations. Most of all, it allows for intuitive, fast and continuous parameter changes reflected in the resulting sonic material. So when considering the design of a music performance system, the spatial principle seemed to be a productive path since the beginning. But no matter how inspiring, in most of the existing tools also serious constraints limiting their performance use as a complete instrument could be observed––although this does not in any way exclude their potential compositional contribution or partial usability on stage. First of all, some of them (*space-master* patch, *PresetInterpolator* or *Influx* classes) can be very helpful albeit they provide just the basic intepolation funcionality encapsulated in a control object requiring additional code in order to generate sound structures. In the case of more complete instruments, the biggest drawback is that the navigation concept is applied only to one domain: either it affects timbre by controlling the synthesis parameters, or it deals with temporal sequence morphing of symbolically represented musical structures, i.e. rhythm and melody. Furthermore, even in the case of interactive interpolation of temporal sequences (e.g. *DMorph)*, the input material is based on fixed pieces of existing music in the form of MIDI files and is limited just to a couple of simultaneous morphing sources. The only instrument which took a less rigid path was the *Gestrument* which introduced certain degree of randomness to the note sequencing, but, on the other side, with a lacking timbral flexibility. Further sonic limitations of the commercially available tools are based on the fact that they are usually bound to a particular sound engine, with the interpolation option serving only as an extension of its functionality. In regard to their input capabilities, most of the tools are two-dimensional, which is another factor constraining the potential expressiveness.

Based on the aforementioned observations it seemed like a tempting idea to design a system that would take the advantage of the principle of spatial navigation in the parameter spaces, but––referring to my ideas in the chapter about material organisation––it would allow for more complex creation of dynamic sonic morphologies and offer several additional layers of agency. This would necessarily imply the integration of both the timbral and temporal domain with the use of generative algorithms capable of shaping, variation and evolution of the material based on the performer's gestural input. Building these ideas I envisioned a performance system combining several layers of control and fostering various modes of creative thinking: convergent, divergent, tacit and analytical. This idea resulted in the merging of various features into one powerful and flexible engine. To rephrase the desired system requirements mentioned previously, its seminal features are following:

- A direct access to a desired sonic **behaviour**

- Continuous morphing of the **timbral** qualities, **articulation** settings and individual **parameter modulations** by gestural movement

- Continuous interpolation of **event** generation settings by the input gestures

- **Generative event sequencing** with algorithms of choice: deterministic, stochastic, chaotic or evolving in time, as well as **responsive** to various control inputs

- **Variable degree** of **behavioural autonomy** – from a **direct gestural control** by the performer through **automation and randomisation** of certain processes (animated trajectories through the *control space* or random evolution of parameters) up to interaction with more complex behavioural objects, including **artificial agents**

Although general interaction ideas were preceding the actual design phase, the actual construction of the system was not a top-down process but rather grew organically, based on testing and experimenting with the various ideas translated into the hardware and software configurations in a bricolage fashion. Among the many considerations that had to be taken during this journey, I will introduce at least the most important decisions related to the key attributes of the system and their realisation.

5.2.1 Reactivity versus Complexity

As already mentioned, in the design of music performance systems there is ususually an inherent contradiction between the intimacy and expressiveness of the system in relation to the performer's input, and the degree of detailed control over many parameters (Dagleish 2013, 69). This problem can be partly solved by automation of selected low-level tasks, but at the same time it is beneficial to keep a holistic influence on the processes. In the previous text I tried to demonstrate how a low-dimensional abstract spatial layout could encourage exploratory and intuitive approaches to the possibilities of sound generation. However, to apply these ideas to a full-fledged system allowing for a rich improvisatory interaction, many decisions and design trade-offs had to be made. One of them was realised in the concept of joining the temporal and timbral attributes into a complex behaviour description that could be addressed by one coordinate in the high-dimensional parameter space. The advantage of this approach is that it supports the spatial aspect of thinking, i.e. involving a cognitive map of accessible locations and trajectories in an abstract space. This principle also suggests that complex sonic changes and transitions can be made through one physical gesture, translated into much more refined and sophisticated sonic process. On the other hand side, an obvious drawback would be the diminished flexibility in being able to address the timbral and temporal behaviours as separate entities. I tried to partly overcome this constrain by adding a control of global parameters influencing the generative processes, in addition to the settings determined by the gestural input.

5.2.2 Variable Degree of Autonomy

Coming back to the notions of agency in interactive performance systems investigated in the Chapter 2, the envisioned improvisatory flexibility of the system implied an implementation of several automation layers controlling the generative algorithms. A vision of being able to switch between a purely "manual" gestural operation and a more sophisticated mode of interaction with generative processes through the changes in higher level parameters led me to the exploration of a rapidly progressing research field focusing on the use of artificial neural networks for creative and musical purposes. In order to introduce a more autonomous and sophisticated behaviour in the system I had been considering several possibilities for the integration of deep learning models such as using variational autoencoders for the variation of sequences (Roberts, Engel, and Eck 2017) or recurrent neural networks for pattern predictions (Avola et al. 2018). But finally, the discovery of an network architecture type called Mixture Density Recurrent Neural Network capable of learning and predicting time-based gestures (Martin and Torresen 2019) was very well fitting with the general idea of spatial navigation in the *control spaces*. Therefore I chose to implement this model as an agent in the system, capable of both an autonomous operation and an interaction with the performer.

Although it can be said that the proposed system interaction options do not involve any agents with high-level cognitive capabilities (Tatar and Pasquier 2019), due to the various levels of automation and gestural interaction they offer complex performance possibilities with multiple degrees of system autonomy selectable in real-time.

5.2.3 Multimodal Approach to Performance

In order to leverage the benefits of different modes of creative thinking in the performance time, it is convenient to have several modes of interaction at disposal, with some of the options facilitating the fast, embodied, immediate and gestural approach, complemented by another analytical and more focused, although slower choices (Tubb and Dixon 2014). Since the proposed system has the nature of a “composed instrument” (Schnell and Battier 2002) with a virtually unlimited number of components, I decided to implement a combination of several different input interfaces to create a multi-dimensional *input space* with several modalities of operation related to the different aspects of a creative process:

Many electronic musicians rely heavily on technology as an extension of their artistic thought processes. If creativity does indeed involve rapid alternation between idea creation and idea selection, systems should be designed with this fact in mind. Divergent or convergent features on their own may be less effective than a well integrated combination of the two. (Tubb and Dixon 2014, 32)

Especially the embodied and expressive gestural interaction part, aimed at the navigation of the *control parameter space* was in need of an appropriate interface. In a somewhat similarly oriented search Tubb and Dixon mentioned several important properties (Ibid., 25) that matched with my list of requirements for such a device: low latency response, high sensing precision, revisitability (possibility of returning to a particular point), relative low dimensionality corresponding to that of the control space, as well as smoothness and continuity (ability of continuous trajectories between the points), and the possibility of––ideally––an immediate access to a certain point in space. I would also add robustness in terms of reliability, and connectability as important features.

Although the sensors capturing motion in a three-dimensional space do exist (such as infrared depth sensing cameras Leap Motion or Kinect), the main issues with these are precision and revisitability, as well as impossibility of reaching a certain point without passing through other destinations in the space. Because of these attributes, a tactile interface was more preferable in the end. I had also considered an option of designing such a sensor myself but because of a recent availability of several controllers with specifications getting close to the requested features I decided to focus on the design of the software system components instead. In the first few iterations I tried to interact with the system via touchscreen interfaces (iPad and iPhone), for which I designed templates in the Liine

Lemur[41] environment. Although these interfaces offered a valuable option of visual feedback, they proved to be either too small (in case of the phone), which meant a low resolution compared to the size of the *control space,* or, on the other hand, lacking pressure sensitivity. This issue seemed to be solved in the Roli Lightpad Block[42] interfaces which offered X/Y and pressure sensing combined with a visual LED indication matrix, but again, the surface was too small and the pressure sensitivity too low for a really flawless interaction experience. So after many trials I finally decided to implement the gestural sensing part with a relatively novel sensor surface Sensel Morph (shown in the Figure 33),[43] offering a good connectivity, reliability and high precision pressure sensing with several ways of obtaining the data: this is possible either through an available force map or via detailed features extracted for individual contact points.[44] The sensor offers no visualisation of e.g. the locations of the points in space, but I have realised this is actually not necessary as the most important is the auditory feedback. I ended up using two Sensel Morph sensors for the gestural input from both hands during a performance, combined with a foot pedal adding an additional control axis, which seems to be a convenient combination of interfaces.

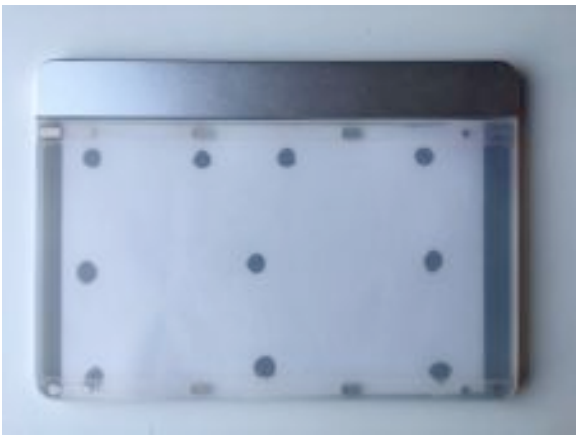

Fig. 33. The Sensel Morph interface with a drawing of a map with interpolation points.

[41] https://liine.net/en/products/lemur/
[42] https://roli.com/products/blocks/lightpad-m
[43] https://sensel.com
[44] As a side note, I am certainly aware of the ideological and market-driven contexts of commercially available interfaces. While I agree that there is politics embedded in the interface part of the creative music making, I chose not to emphasize this fact in the project and instead focus on the very possibilities of interaction in the limited time period.

To get back to the other part of the interaction with the system's algorithms, it is the one that encourages a more analytical and rational approach. This includes more precise and rational operations related to the development of the improvisation on a macro time scale, such as working with memory (storing and recalling settings), activating and deactivating processes, swapping behaviours in various layers (e.g. patterns, scenes or synthesizer programs). For this purpose a different manner of control was necessary which was satisfied in two ways. Firstly, I chose to use an external controller with buttons, pads or switches to enable fast discrete operations in real-time. A second option was from the beginning the actual textual programming interface, i.e. the computer keyboard and display that offes the deepest possible of interaction , meaning changes in the software code:

"Code as interface" allows radical intervention and reconfiguration of musical systems while they are running. On the one hand, it becomes possible to do many previously unimaginable things; on the other, it is often not possible to do them very quickly. (Wilson et al. 2014, 54)

To sum up, as a result of this desicion making process the system can be controlled through the activity in several *input spaces* constituted by the input data. These inputs can be used simultaneosly or sequentially depending on the situation.

5.2.4 Open-Ended Design and Modularity

From the beginning of the design process it was evident for me that the proposed hardware-software assemblage should be as open as possible for experimenting with different input and output modalities, with a stable core algorithm interfacing with flexible peripherials. I understand it as a truly experimental system in flux as foreseen by Rheinberger,[45] always capable of offering new discoveries and surprising novel solutions. Each software or hardware component added to the system considerably extends the creative possibilities but also in a way prescribes certain way of working. Yet some of the design choices were influenced by the opportunities offered by existing platforms or by their limitations. For instance, despite my effort to bring all the components down to a unified software platform––which would simplify the setting up and possibly increase the system's stability––, due to the practical and convenience reasons such as availability of different libraries or lack of connectivity I ended up combining several components written in different programming languages. At first I started to implement the core of the system in *Max* visual programming

[45] As introduced in the Chapter 1.

environment but soon I realised its limitations such as slow and buggy interpolation options as well as certain rigidity of the structures. Despite being a relatively new and intense learning experience and challenge for me, the *SuperCollider* environment has proven to be a much more flexible design option considering the powerful generative sequencing options, more logical and clear structure of the code and the possibilities of real-time changes in the algorithms. Furthermore, for practical reasons of speed and reliability I decided to implement the feature extraction of the sensor data from the *Sensel Morph* as a separate *C++* utility, whilst the neural network model used for gesture prediction is accessed through a *Python* script. All these software components communicate via OSC (Open Sound Control) protocol.

5.2.5 Symbolic and Sonic Output

Since the core of the system essentially offers a universal method for the control of sonic behaviours, I realised that it would be beneficial for the various use cases if the system would have built-in several options of signal generation, meaning both both symbolic instructions (MIDI), control voltage signal (CV) and audio itself.

The reason for the usage of a somewhat outdated and inflexible MIDI communication protocol lies in my interest in expanding the possibilities of the control of existing hardware instruments while preserving their iconic timbral character. I was always curious how it would be possible to overcome the limitations of the classical interfaces to be able to access the instruments' timbral spaces in a more fluid manner.

The output of control voltage (CV) signal is another interesting option that can be used to interface to a different types of hardware synthesizers equipped with this option. This is essential for building sonically powerful hybrid analogue-digital systems with modular synthesizers that are offering much potential: e.g. distinct sound quality of analogue VCOs, but also a different workflow. To be prepare for this option (although currently not implemented in full), an output of control voltage from the system has been made possible through the use of several interfaces (essentially digital/anaglog converters): ES-8, ES-3 and ES-5 by Expert Sleepers, offering in total 16 audio or CV outputs plus 5 so-called gate outputs for triggering events.

Since as I have already mentioned, the system's core is implemented in the *SuperCollider* environment offering not only sequencing but, most of all, huge sound synthesis and processing capabilities, an obvious choice is to use its internal sound engine. I tend to use the internal *SuperCollider's* native Unit Generators for sound creation as a complementary option, but so far my main focus was on the exploration of the timbral possibilities of available VSTs (Virtual Software Instruments) for the reason of a similar parameter structure and MIDI communication options with the hardware synthesizers. In this way most of the system's core components could be used easily with both the virtual and hardware synthesizers. In the first months of the project development the only option to realise such strategy was to use yet some another application that could host VST plugins and receive MIDI control data from *SuperCollider*. I successfuly managed to set up such a configuration with the *Ableton Live* program. But since quite recently it is also possible to host VST plugins in *SuperCollider* itself, I decided to switch to this option for the reason of convenience. The biggest advantage of the "VST approach" is the combination of the complete synthesizer designs with the ability of an advanced control such as a complex modulation of the parameters. On the other hand it is plain enough that the ready-made software instruments are far less flexible in their sonic possibilities––which makes them, however, not less interesting for the sake of timbral explorations.

5.2.6 Visual Feedback

After several months of considerations regarding a visual feedback during the performance I realised that it causes too much distraction that keeps me from fully concentrating on the sound processes. Therefore I decided to keep the graphical user interface very minimal and only optional in order to primarily focus on the auditory component. As Joel Ryan notes:

> *The horror of visualization is partly avoidance of the 'extra musical' but there is also a real fear of the dominance of the one mode of experience over the other. While always involving some visual references, musical instrument design for me seems to be all about trying not to clutter up the interface with visual tasks which crowd out listening. (Ryan 2002)*

He then continues with an argument that "*difficult bodily involvement with playing don't seem to interfere with concentration on sound, perhaps the opposite*", which has proven to be valid as well for the gestural part of my interfacing with the system.

Following on these ideas and considerations, in the following section I will introduce the concept and architecture of my performance system.

5.3 Concept and Architecture

AMEN – The AMbiguity ENgine[46] is a system aimed at creating and shaping dynamic sound morphologies and "fluctuting mesostructures" by exploration of parameter spaces allowing for morphing of sonic identities. It uses generative algorithms and artificial agents and it is meant for use in real-time improvisatory situations, both solo and with other musicians.

AMEN is a "composed instrument" in the sense that the interface, control software and sound engine communicate through OSC or MIDI messages that allow the decoupling of the components, although they can be also combined in one physical device such as a laptop computer. So, the system is an experimental setup with a partly modular hardware configuration. The software part has also a flexible architecture to a large extent: it allows for some code elements to be replaced or changed on the fly in performance time, e.g. by the execution of code snippets.

In its current stage, the main input interface consists a commercially available, high-definition pressure-sensitive multitouch sensor surface used for the gestural input, which allows for continuous navigation in a three-dimensional space. A second sensor is used for controlling additional parameters of the generative processes with the other hand.

Another input devices such as a pressure-sensitive grid controller can be employed to gain more control over the discrete settings of the system (e.g. changing synthesizer setups), meta-level control and automation (switching "agents" and automated processes on and off), long-term memory access (saving and loading "states" or scenes).

Soundwise the system is generally very flexible in that it is "synthesis-agnostic" and can be, with some ajustments, used for the control of various types of sound synthesis. This includes the internal *SuperCollider* engine and various virtual software instruments (VST), but also external hardware synthesizers capable of MIDI communitation, or voltage-controlled synthesizers as the modular systems in the Eurorack standard.

[46] The name was inspired by Joel Ryan who mentioned ambiguity engine in one of the conversations we had at STEIM. As is well known, "amen" also means "so be it", which seems like an appropriate reflection of the improvisatory practice with generative algorithms, at least in some situations.

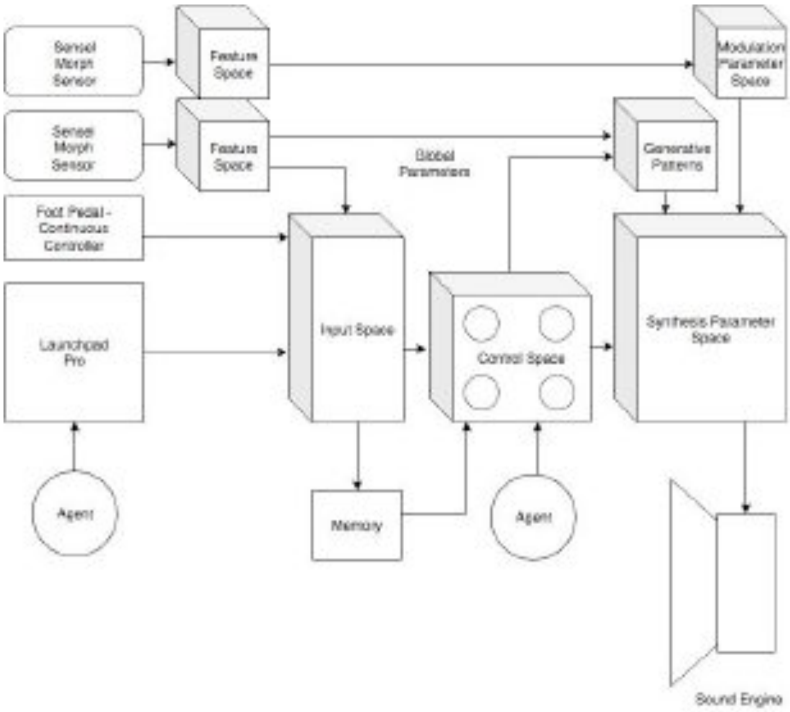

Fig. 34. AMEN System Architecture.

5.3.1 Input Feature Space

The *input feature space* is a space of data acquired from several connected hardware interfaces. It consists of the computational *gesture space* defined by the motion data extracted from two identical pressure sensor surfaces, and adjacent *feature space* layers containing higher-level features extracted from the gestures (surface area covered by an object such as a finger tip, whole area between several contact points, maximum distance between the contact points, global pressure, time deltas describing the speed of movement in each of the three directions). Other input data comprise the continuous control signal from a foot pedal and data from a grid-based controller equipped with sixty-four pressure sensitive pads.

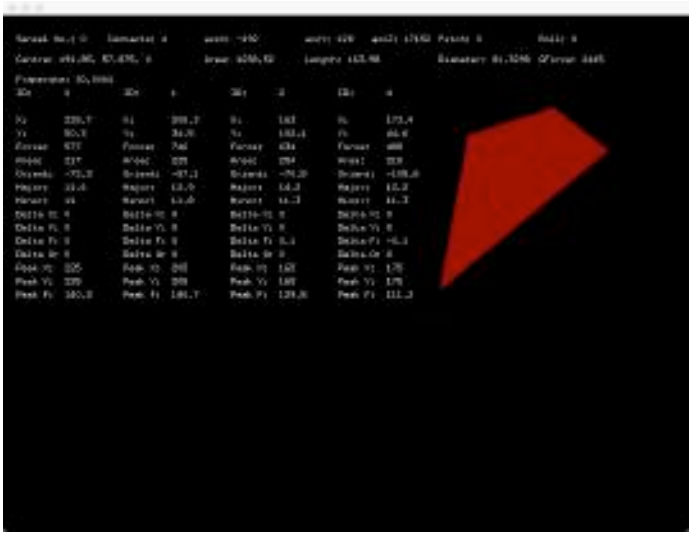

Fig. 35. A screenshot of a small utility application programmed in OpenFrameworks/C++ for the extraction of features from the Sensel Morph sensor. The shape on the right depicts the position of the fingers on the surface.

5.3.2 Core

The software architecture of the system can be represented by three interconneced arenas of computational spaces: the *input feature space*, *control space* including several layers and the *parameter space,* again with different parameter subsets. The heart of the system is a three-dimensional *interpolation control space*. This space can be filled with a reasonable amount of points called **Locations**, containing the control data for the the generation of algorithmic **Behaviours**. The **Behaviours** (on the computational side) are in turn responsible for the generation of dynamic sound morphologies (on the perceptual side), associated with a particular location. So, whenever the cursor is moved to a particular **Location** in the *interpolation space*, respective parameters describing a **Behaviour** will be continuously sent to a generative algorithm. If cursor gets positioned between several **Locations**, the control data will be interpolated according to the proximity to the nearest **Locations** which results in a mixed or “inbetween” behaviour, in both timbral and temporal sense.

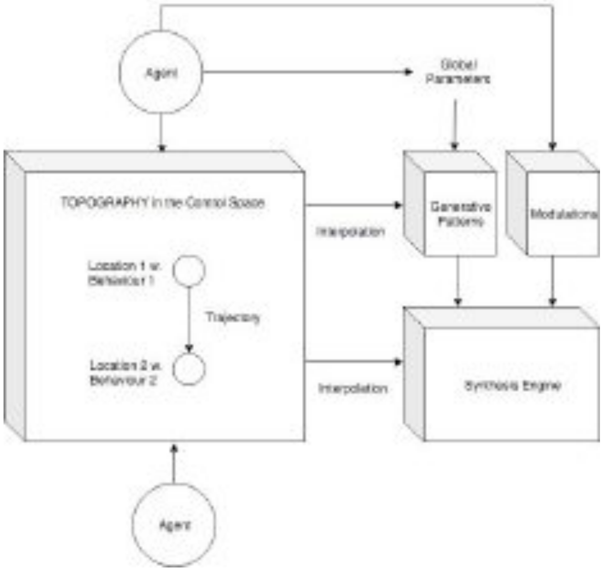

Fig. 36. Diagram of Topographies, Locations, Behaviours and Agents in the AMEN system.

5.3.3 Control Space

The main control space is an *interpolation space* with three dimensions. It is virtually unlimited in size but the main area is where the **Locations** with data for the generation of **Behaviours** can be positioned.

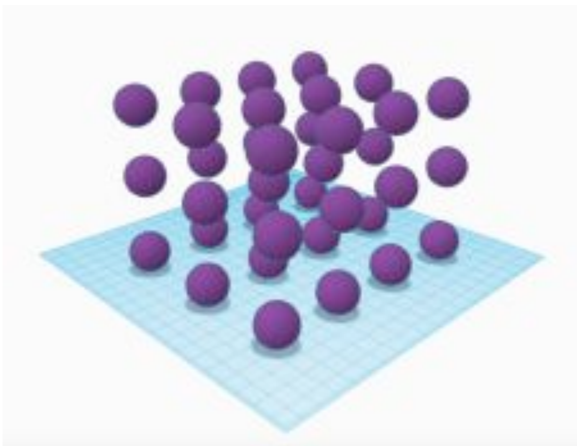

Fig. 37. The interpolation control space.

5.3.4 Locations

The **Locations** can be arbitrarily positioned in the control space by specifying their respective X, Y and Z coordinates. I chose to organise them on three horizontal planes. The reason for this was based on the practical possibility of remembering the positions and also to be able to access them via a primarily two-dimensional input controller allowing for precise gestural movements on the planes.[47]

5.3.5 Topographies

A topography describes a constellation of all the currently loaded **Locations,** meaning their coordinates in the *control space*. It can be saved and recalled as a scene preset independent of the **Behaviours**, although most of the time they are used in combination, e.g. during the initialisation of the setup. However, the separation allows for changes in the topography even while the same **Behaviours** are loaded, which leads to different sonic results resulting from the same gestural movements. Commonly, a **Topography** consists of up to twenty **Locations** with stored **Behaviours,** arrange on two layered planes. This seems like a reasonable amount of distinct points given the size of the sensing surface and a cognitive ability to remember (and also to design) the **Behaviours**.

Fig. 38. The system screen with optional GUI elements: the PresetInterpolator window showing the Topographies (in the centre: 2D projection of a 3D interpolation control space), synthesis control parameters (top), some randomly generated global parameters (right) and the live coding textual interface (left).

[47] In a more recent version I am utilising only two planes which makes the space more easily accessible.

5.3.6 Cursor

A **Cursor** is represented by a vector describing the position in the 3D *control space*. The **Cursor** position and movement can be controlled by various data streams, originating in the *input spaces* of hardware interfaces or elsewhere, such as generated by an artificial agent. The **Cursor** position gets translated into the particular weights defining the amount in which the respective **Behaviours** participate in the sound generation.

5.3.7 Behaviours

> *The term behaviour satisfies the need we have to capture a continuum between the active qualities associated with software such as live algorithms to the more passive characteristic of a simple software synth, all of which can be described as possessing behaviour in a musical sense. (Bown, Eldridge, and McCormack 2009, 193)*

Behaviours, or more precisely, behaviour definitions represent a position in a high-dimensional parameter space defining the properties of timbre and time varying articulation and event generation (i.e. rhythmical pulsation and relative pitch: even in the case of non-pitched sounds, they are triggered by events containing notes with pitch information). These are basically parameter settings that are responsible for the generative process producing the sound objects. The settings can be manually crafted and fine-tuned, but also generated algorithmically and experimented with outside of performance time.

A **Behaviour** definition is stored in a data object. When ”performed” (i.e. when patterns are being played) it generates data and sends it to the syntesizer engine where it generates sound. Since, due to the design of the system the synthesis engines or their internal settings can be changed independently, the same behaviour definitions can cause different sonic results based on the connected objects.

A **Behaviour** is composed of several superimposed data layers containing subsets of parameters for the synthesis and sequencing of events:

Layer 1: Sound synthesis. Synthesis parameters and their number vary depending on the engine. They define the timbral properties of a sound object, including possible "internal motion" of the sound on meso and sub-meso-level (in case of a complex synthesizer structure with several low-frequency oscillators, envelopes or other internal modulation sources).

Layer 2: Event pattern definitions at each Location are responsible for the timing, velocity and relative pitch of the emitted sound objects, they esentially sequence the "note" events. Patterns describe rules for event generation in a stochastic manner, all influenced by global high-level control parameters described below. Various probability distributions can be chosen in the particular patterns for the local implementation of the global parameter settings.

Layer 3: Articulation pattern settings for the amplitude envelopes of each **Behaviour**, defining the overall shapes of the sound objects and their evolution in time by stochastic procedures.

Layer 4: An additional layer of **modulation settings** describes the parameters of external modulations applied individually to the each of sound synthesis parameters. Modulations can be done with control-rate signals generated by dedicated "control synths", if the internal sound engine or external voltage-controlled engine is used. In case of external hardware or software accepting MIDI streams, modulation is done with MIDI generating patterns. The modulation signal can be also routed to control other modulation synths, thus creating a complex behaviour. As of now, the modulation layer is actually global and is not assigned to the particular **Behaviours**. This allows for creation of global modulations regardless of the current **State,** e.g. by an external data input.

So far, several modulation types have been tested and proven useful, with many more options to be possibly added in the future:

- Low frequency oscillators causing slow gradual changes in several synthesis parameters.

- Virtual connected spring oscillators with different settings for each synthesis parameter, excited by an external input (tapping on the pressure sensor).

- Random walks for each parameter – causing constant "drifting" of the sound, with the option to return to the original parameter constellation.

5.3.8 Global Parameters

The behaviours are interactive in that they respond to changes of few high-level parameters. Settings of these parameters affect the attributes of the produced sound events.[48] The ranges for the parameters (apart from Transpose) are standardised to be from 0 to 1.

- ***Loudness.*** A coeficient defining the overall volume (all sound amplitudes are multiplied by this).

- ***Speed*** – Speed of the event generation. This is a relative event frequency depending on the overall tempo and the particular pattern definitions.

- ***Density*** – Temporal density describing the probability of an event appearing at each step. If the density is set to 1, events will be emitted always as prescribed by the pattern, 0 means no audible events at all times.

- ***Entropy*** – Maximum of temporal deviation from the quantized rhythmical grid, defined proportionally to event time deltas.

- ***Pitch deviation –*** Maximal deviation from the pitch prescribed in the original pattern definition. The pitch information will be adjusted in a probabilistic fashion based on this value.

- ***Envelope deviation –*** Affects the maximum variation in the amplitude envelope shape parameters (attack, sustain and release of the sounds).

- ***Notes –*** Definition of one or more main pitch centres as a reference for patterns. An array of several notes results in simultaneous events with different "pitches".

- ***Transpose*** – Transposition parameter controlling for the pitch centre.

How does it work in practice? For instance, I can decide to have a series of short, "high-pitched" sounds emitted at variable rate, with accelerando and subsequent ritardando stored at a particular **Location.** The temporal character of the behaviour will be prescribed (with certain probabilistic freedom) through the event and articulation patterns (LAYER 2), with a particular timbre defined by

[48] The inspiration for this type of control comes from Curtis Roads' notion of oppositions as discussed in Chapter 2 and from Bjarni Gunnarsson's concept of the so-called polarities (Gunnarsson 2012, 40). These are used as attribute names for the global control settings of for all code objects responsible for the sound generation, albeit with different internal implementation or different weights for each of them in the object-oriented programming fashion, referring to a feature of called polymorphism.

a configuration of synthesis parameters (LAYER 1). The behaviour is furthermore influenced by the current configuration of global parameters and the actual modulation signals. Based on the current **Cursor** position (setting the weights for each behaviour) the Behaviour contributes to the attributes of the finally generated sonic structures.

5.3.9 States

A state represents the momentary setttings of all sound generation processes, including all the loaded **Behaviours** with the patterns and synthesis parameters, the current settings of the interpolation weights for each **Behaviour** (equivalent to the cursor position) and global parameters, as well as parameters influenced by current external input from the sensors. A state can be stored, recalled, saved to disk and transformed by further operations (e.g. sequencing or evolution of the parameters).

5.3.10 Scenes

A scene is a construct consisting of the configuration descriptions of several objects. It is the top-level setting of the system, including the currently loaded **Topographies** with the **Locations** s **Behaviour** definitions, as well as global parameter settings.

5.2.11 Trajectories

Trajectories are understood as abstract gestures in the multi-dimensional parameter spaces and belong to the most important control concepts. They can result from the movements of **Cursors** in one or several *interpolation control spaces*, or from other types of movement in the various parameter (sub)spaces. In the former case, a **Cursor** can be moved in multiple ways depending on the current mapping settings that connect it with an actor or agent, which can be any computational or real-world object capable of agency. The system can generate and respond to several kinds of trajectories:

1. "Manual". Trajectories derived from real-world gestures produced by human or non-human agents in the physical space, acquired from a stream of data in the *input feature space*. This can be a translation of performer's physical gestures captured by a combination of available input sensors (pressure-sensitive x/y surface, pedal, buttons), but also of input data produced by various physical objects either moving or standing still on the sensor surface (metal balls or other different-shaped objects), as well as animals or robotic agents (e.g. a robotic ball). In practice I am mostly using my own hands with pressure sensors and my foot with a pedal controller, but I also sometimes position objects on the sensor surface which adds another possibilities of interaction with the system.

2. Automated. Prescribed **Trajectories** of a certain shape with a duration, describing a motion from one point to another can be realised either in the *interpolation control space*––by designing an animation of the **Cursor**, or in other parameter spaces. The system allows to execute a command realising the motion from one **State** to another stored **State** in a certain time period. This option is also accessible for triggering by the external grid-based controller, with pads representing the available **States**. In this way it is possible to create a transition between two known **States**, or a transition from a current state (e.g. as defined by a present input from the interfaces) to a known stored **State**. Contrary to the physical hand gestures, the automation with exact settings allows me to achieve precise transitions where intermediary moments of uncertainty finally resolve into a stable sonic behaviour. Besides the movement in the control space, another available option is to trigger an animated **Trajectory** leading form a certain point in the *synthesis parameter space* to a different one instead. This process results in a straight-forward timbral morphing (technically done by a gradual linear interpolation or "crossfading" of parameters), thus bypassing the control space with its complex influences on event sequencing processes. As of present, only a linear motion is implemented as it has proven to be the most clearly perceivable and convenient way of generating musical transitions.

3. Sequenced. Multiple animated **Trajectories** can be chained and sequenced in time, with the application of random procedures for the selection of points from a pool of available options. In this way, an automatic traveller agent can be set up to traverse the space between the stored **States**, e.g. with random transition times.

4. Autonomous. This type of **Trajectories** is derived from a movement of the cursor in the *control space*, that is mapped to an autonomous agent. I am describing the types of agents in the folowing passage.

5.3.12 Agents

Going back to the idea of behavioural objects, both (or more) of these sound processes can be considered as agents with different properties and degrees of autonomy. They can be programmed to generate various types of behaviour influencing the generation of sonic objects, ranging from periodic to irregular and chaotic behaviours. In order to achieve large potential complexity of interaction and sonic evolution, the system enables multiple levels of interaction between objects of different agency. Several types of agents can interact with and within the system:

1. Real World Agents. As I have already stated, the interface allows various real-world agents to interact with the algorithms. While pads on the grid controller are essentially operated by me as a human performer, the pressure surface sensor is sensitive enough to enable sensing og other organic or inorganic physical objects.

2. Software Agents. To achieve a greater degree of sonic complexity beyond manual control of the performer, several software agents can be triggered to either influence the evolution of the current **State** in a probabilistic way, by changing the global control parameters, to automatically navigate the *control space,* or to generate additional accompanying synthesis voices derived from the control data created by the main event pattern.

While these behavioural objects could be called reactive agents, an additional autonomous agent can be activated to traverse the control space by itself or in response to the gestural data from the input space. This agent generates 5-dimensional gestural motion data (3D spatial coordinates, pressure data and time deltas) mapped to control the **Cursor** position in the *interpolation control space.* It is an artificial neural network model (Mixture Density Recurrent Neural Network) trained on the gestural data recorded during my previous performances. Two modes of interaction with the agent can be selected from: while in the "Call and Response" mode it reacts to the performer's gestures immediately after they stop, the "Battle" mode allows for simultaneous gestures generated by both the performer and the reacting neural network model. The latter case makes sense when an additional *interpolation control space* is employed, which can be a data space similar to the main *control space* but mapped to a different layer consisting of another synthesis "voice" or sound generation algorithm.[49] The interaction mode settings can be switched in real-time, while it is also possible to adjust the parameters influencing the degree of randomness in the gestural behaviour.

[49] In addition to this obvious option, other possibilities are imaginable that have not been tested thoroughly, e.g. to map the agent's gestures to the global control parameters, or to the separate the control of timbre and event generation and divide the tasks to be served by either human or the neural network each.

More technical details about the agent are provided in the Technical Implementation section.

5.4 Operations, Interactions, and Interventions

In order to further prepare the system for a real-time use, some operations outside of performance time have to be undertaken. Algorithmic **Behaviour** definitions have to be constructed in advance, assigned to **Locations** and saved as **Topographies.** This process comprises the design of patterns for generative event sequencing with individual settings of the respective boundaries and response coefficients/weights in regard to the global control parameters, as well as the choice of synthesis engine and its settings. This creative phase can be quite time-consuming, so I use various helper tools such as a parameter randomization algorithm that can help to speed up the process of preparation and can generate sufficiently interesting, varied and large database of the **Behaviour** configurations.

The system architecture described in the previous section defines a “battlefield” that will be used for real-time operations in a performance context. In the following passage I will focus on the actions that can currently take place in this possibility space. While some of the real-time actions are assigned to physical controllers for immediate use, others are accessible via modification and execution of prepared text-based commands, which constitutes an additional control and intervention option during the performance.

5.4.1 Gestural Interaction

The core system operation is represented by the embodied gestural interaction of the performer via one of the two pressure sensitive sensor surfaces that allows to trigger and modify sound generating algorithms by navigation in the *interpolation control space*. The position of an object touching the interface is translated into **Cursor** position in the *control space*. In addition to that, other features extracted from the input data are mapped to the global event generation parameters. In this manner, exploratory dynamic morphing of sonic behaviours can be achieved through an intuitive and responsive multitouch interaction. In this mode, the sound is only produced when an object touches the surface, with touch pressure influencing the overall volume. The other identical surface is used to influence synthesis parameter modulations and control transformation operations such as repetition of events.

5.4.2 Activation of Processes

The described gestural interaction is a very fluid and responsive method of approaching the sound generation on the meso timescale, and also when quick and instant performer's reactions in an improvisatory context are desirable. But in order to be able to create a development of temporal structures on a larger macro time scale, reaching to several tens of seconds and longer––for instance in a solo performance––some additional control operations can be thought as very useful. Therefore the system offers a palette of actions that extend the possible ways of interaction beyond the immediacy of the gestural input. These include the activation and deactivation of automated processes and software **Agents**, assigning stored **Behaviours** to **Locations,** changing the **Topography** of the *control space* by moving the **Locations,** or reloading the whole **Topographies** from disk**.** These operations can be realised with a grid-based pad controller (storing, recalling and morphing between **States**), through an additional touchscreen tablet (visualisation and control of the **Locations** and other settings)**,** or via execution of text-based programming commands.

5.4.3 Repetition and Variation

One of the very useful options when improvising with algorithms constantly generating novel sounds is to be able to get back to the sonic material that appeared in the past. The system offers to work with both the short-term memory––which is made possible by immediate repetition done by looping of recent sound events and their subsequent transformations (e.g. changing length of the loop or timbre settings) ––, or with long-term memory by storing and recalling **States** and **Scenes**. This offers a possibility to get back in history and work with the material discovered in the past, including the most valued configurations stored during the past performances. As mentioned previously, it is then possible to create **Trajectories** between various stored **States** or dynamically morph between the current **State** and a chosen past **State.** It is worth noting though that despite reverting to the previous settings as an actual option, the sonic material is never going to be completely identical due to the stochastic nature of the algorithmic **Behaviours**. So technically even when recalling past **States,** the system is going to generate some sort of reminiscence of the original events. Furthermore, for practical reasons the modulation settings are currently not stored with the information describing a **State,** which enables to change or morph between several **States** while keeping the continuity of the modulation signals.

5.4.4 Text-Based Interventions

As mentioned previously in this chapter, the textual interaction––also known as live coding**––**is especially useful for stepping beyond the momentarily available settings and mappings of process parameters to controllers in a performance situation. In this manner, programming commands can be used for injecting new (and untested) behaviours into the system: whole layers or just single algorithmic objects can be changed. The possibilities here are virtually huge, although not free of any risk depending on the level of intervention. But in general it can be experimented with the various modulation signals, randomisation of synthesis or global parameter settings, and many other processes, completely bypassing the other control inputs. The live coding approach is, of course, considerably slower compared to the immediacy of gestural and haptic interaction. However, due to the flexible nature of the software environment it is an intriguing option as it opens new windows into undiscovered territories.

6 Evaluation

6.1 Epistemic Dimension Space

What kind of music performance system is AMEN? In Robert Rowe's terminology, it would obviously a performance-driven generative system, using various rule-based algorithms for the generation of musical structures. In regard to the "performance versus player" continuum the situation is more complicated because of the various performance modes and autonomy of the agents, so I would suggest a view that accents a multifaceted interaction of objects with various degrees of agency.

In reference of different abstract spaces and their connections, Magnusson developed an interesting model of a visual representation of "epistemic features" in an 8-dimensional space with the aim of summing up the properties of interactive music systems (Magnusson 2010), based on the so-called design space analysis approach suggested previously by Birnbaum et al. (Birnbaum et al. 2017).

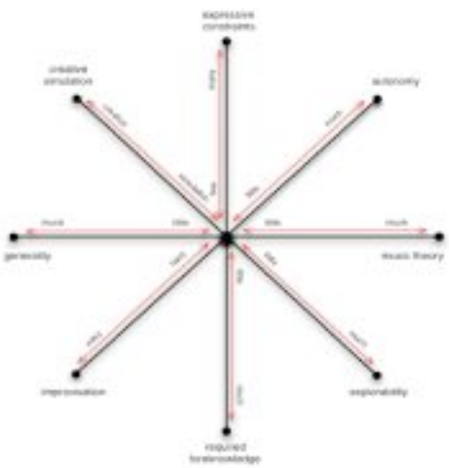

Fig. 39. Epistemic dimension space of musical systems (Magnusson 2010).

According to Magnusson, the visualisation is suited to depict *"parameters that are unique to heavily abstract, conceptualized and symbolically designed musical tools"* (Magnusson 2010, 45), which is why I will now try to use this method to sum up the features of the AMEN system with an explanation of the dimensions. I will quote the original meaning of the dimensions in italics.

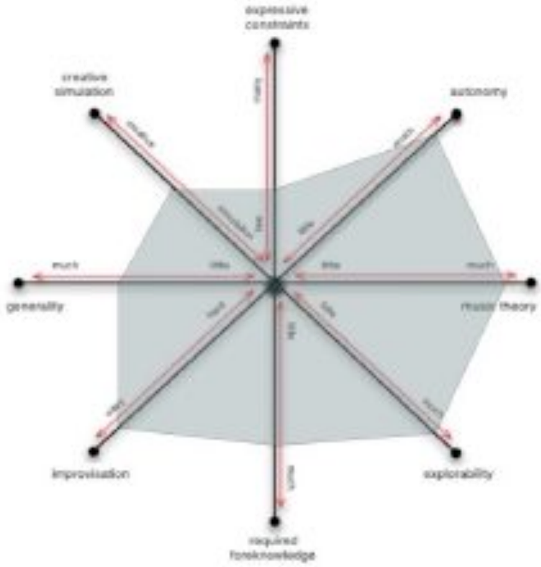

Fig. 40. Epistemic dimension space of the AMEN system.

- ***The Expressive Constraints*** *axis focuses on the expressive limitations outlined by the tool's design. This is the space of musical possibilities.*

It is true that in a single configuration of approximately twenty Behaviours arranged on two horizontal planes there are considerable expressive contraints. The interaction possibilities can get quickly explored when using only the gestural interfaces without changing any additional settins. But based on the possibility of activating various automations and agents as well as the option of morphing between the current and several stored States, in addition to the fact that both the sound engine and generative algorithms can be freely swapped of modified in real-time, the possibilities vastly increase.

- *The **Autonomy** axis specifies the degree to which the instrument provides the functionality of an automata. Certain musical tasks are delegated to the instrument(often using artificial intelligence), possibly responding to a performer.*

Although I managed to integrate a complex neural network model into the system's architecture, the "intelligence" or autonomy of the system as a whole could be still disputable. The agent interaction is certainly enriching the performance possibilities but since as of present the model is in fact "deaf" and only reacting to the performer's gestures, its behaviour feels somewhat random. In terms of the "meaningfulness" of the interaction, it remains to be explored if a further training of the model on more gestures, or some kind of connection with a listening agent acting base on the real-time sonic attributes would contribute to a more sophisticated result. Another interesting question remains open, wheter the computational space of gestural interaction with the network model should be shared or separated in terms of influence on the shaping of the sound objects. Anyway, it is in many ways interesting for me as a performer to be able to switch the roles––by attributing more agency to the system I can become an observer with the possibility of intervening in the process.

- *The **Music Theory** axis represents the amount of culturally specific music theory encapsulated in the instrument, in terms of the possibilities for various tonal and rhythmical structures, as well as signal processing. This could typically be scales, chords, arpeggios, or time signatures.*

In order for the system to be prepared for a performance, the generative algorithms of the Behaviours have to be crafted in advance, which requires a considerable knowledge of theory and programming.

- *The **Explorability** axis represents how much depth the instrument holds. This factor is critical with regards to how engaging the instrument is and affects learning curve and the possibility of flow.*

There is certainly much to explore in regard to the different interaction modes and influencing the generative algorithm settings, as well as memory operations during the performance.

- *The* ***Required Foreknowledge*** *axis represents the fact that many systems do not require much musical knowledge in their design or performance as they contain it already.*

In order to make full use of the system's capabilities, this would also include live coding interventions implying the need of the knowledge of the *SuperCollider* language. However, since a big part of the system's functionality accesible through the external physical interfaces, it is also possible to be used without an extensive prior knowledge.

- *The* ***Improvisation*** *axis indicates the degree to which the instrument lends itself to free improvisation. How responsive is it, how open for changes in real time performance and how quickly can it be adapted to those?*

Since I have designed the system with an improvisational setting in mind, it is well suited for this kind of use.

- *The* ***Generality*** *axis denotes how open in expression the instrument is and how well it copes with the multiplicity different of musical situations.*

As mentioned before, the system has been tested in solo and collective improvisational situations. However, in regard to its open-ended character, it is necessary to do adjustments of the algorithms and synthesizer engines to prepare it for different musical situations.

- *The* ***Creative-Simulation*** *axis captures whether the instrument is novel in terms of interaction, sound and function or an imitation of established tools and practices.*

The project extends the well established principles of navigation in parameter spaces to a full-fledged performance-oriented system which utilizes timbral morphing and generative sequencing abilities, together with integration of more or less autonomous agents traversing the spaces.

6.2 Reflections

> *If a computer program is written by the composer, the development of the program is an integral part of the compositional process (since ultimately it is driven by the same motivations). Since the program is designed solely for use by its developer, there are no methodological constraints placed on its construction. Furthermore, there is no need to define any rigorous criteria for success nor to use such criteria in evaluating the program and the compositions. If the composer intends the music for public consumption, then they may only be evaluated in the same way that composers and compositions are usually appraised: through audience reactions at performances, record sales, critical reviews and so on.* (Pearce, Meredith, and Wiggins 2002, 126)

The AMEN is an experimental system in constant flux. It is relatively open-ended and universal in terms of connectivity, capable of vastly expanding the sonic possibilities of existing sound generation engines as well as their intuitive real-time control. It fulfils the need for the production of surprising and novel sound morphologies that can be further manipulated and used in improvisatory contexts. Contrary to some other available music production tools utilizing the spatial navigation principle, the system essentially does not serve the purpose of exploring huge parameter spaces. It should instead allow for an expressive way of generating and interacting with dynamic sound morphologies accessible by movement of various types of agents in the control space.

The AMEN system has been gradually evolveding during the past two years and I have tested its various iterations in concert settings, both in solo and duo improvisations. It can be said that it has proven to be a flexible tool for a computer improvisation with many options still waiting to be explored. While in the early design phases the idea of navigation in the control space felt promising but still lacking more embodied real-time control and memory operations, this has significantly improved with the addition of pressure sensitive sensors allowing for a precise control over the dynamics, and the pad-equipped interface for discrete operations such as storing and recalling the States of the system or adding and swapping Behaviours in the Topographies in performance time. Whereas the option of morphing between system's States has been useful in that it allows for continuous sonic transitions and "bulk" parameter adjustments, in order to achieve a suffucient variety, some settings have to be changed directly in a dicrete fashion.

As I have already mentioned, the coupling of timbre characteristics and properties of time sequences stored in different positions in the *control space* with the ability to navigate freely between them can be viewed as both an advantage and a drawback. It can be sometimes desirable to have the option of modifying the one while keeping the other stable. Although this was a design choice from the beginning, an optional separation with separate control of the two layers could be considered in order to enrich the interaction possibilities.

Among the affordancies of the system there is a flexible sonic output, expressiveness in terms of the reponse to performer's gestures, ability to generate surprising and to a certain extent unpredictable sonic material and also intuitive control mediated by the spatial principle. This is true at least for some part of the functionality, but there is a high ceiling for what can be achieved by more advanced interventions such as with the live coding approach.

In regard to the constraints, several technical and design issues can be identified. The size of the interpolation space feels somewhat limiting since it is determined by the physical dimensions of the pressure surface that can be be basically encompassed with one palm. This means that only a handful of Locations can to be loaded at the same time onto one Topography to allow for meaningful morphing and interaction with the Behaviours. This is an interesting design problem that could be maybe solved by some sort of windowing approach in the style of "multiple desktops" offered in some computer operating systems. An occasionally and unpredictably high CPU usage when using VST instruments together with multiple modulation control synths (up to 80 complex oscillators simultaneously) is also a considerable technical limitation which encourages a rethinking of the design––at least some optimisation of the modulation layer would be certainly a way to go. This is also among the main factors constraining the vertical sonic density in the current version of the system. The use or "abuse" of ready-made VST plugins for sound generation could be viewed both as a constraint and an affordance. The system is however open to be extended mainly by the use of native *SuperCollider* engine for sound generation, which it currently applied only to a certain extent.

It has been interesting to observe how the system design encourages particular ways of interaction or styles of "playing" depending on the situation: in this way it crosses the boundaries between the more embodied gestural way of performing with digital instruments and a more detached style resembling an occasional interaction between self-sufficient autonomus objects. Probably the most important factor influencing the interaction style is whether it is a solo performance or improvisation with other people. Unsurprisingly, the solos put more pressure on the variety of material the system should be able to generate, while they also beg for the possbility of additional sonic layers (together with more agents responsible for their generation). On the other hand, the system has proven to be responsive and expressive enough for a collective improvisation context, where it can generate a large variety of surprising yet still controllable outputs.

Due to the open-ended character of the setup and many potential options for generative algorithms or synthesis engines there is a relatively large possibility space in regard to the possibile outcomes. However, these possibilities also bring up the question of time resources, or, in anoother words, question of "breadth" versus "depth": is it better to spend time practicing with the same settings, scenes and parameter configurations, or rather explore completely new sonic territories? To be able to answer this kind of questions I would need more practical experience with the system.

6.3 Conclusion

In my research I have explored strategies of combining generative algorithmic procedures for the creation of dynamic sound morphologies in a music performance system for computer improvisation. The system is based on a concept of spatial navigation in an abstract control parameter space populated with different algorithmic behaviours responsible for the control of sound synthesis and stochastic event sequencing. Performer can interact with AMEN – The AMbiguity ENgine by gestures of his both hands performed on high-resolution pressure sensitive surfaces, as well as by other external input interfaces in a more discrete manner. The system furthermore merges different types of human and non-human agency that can navigate the computational space and thus influence the algorithmic generation of sound objects.

The AMEN system has been tested in various live performance situations and its open-ended design makes it suitable for further extensions. In the future, these could include the employment of various different synthesis engines (control voltage or MIDI based, or *SuperCollider* UGen architectures), as well integration of another artificial neural network models, e.g. for autonomous sequence evolution.

Bibliography

Avola, Danilo, Marco Bernardi, Luigi Cinque, Gian Luca Foresti, and Cristiano Massaroni. 2018. 'Exploiting Recurrent Neural Networks and Leap Motion Controller for Sign Language and Semaphoric Gesture Recognition'. *ArXiv:1803.10435 [Cs]*, March. http://arxiv.org/abs/1803.10435.

Bader, Rolf, ed. 2018. *Springer Handbook of Systematic Musicology*. 1st edition. Springer Handbooks. New York, NY: Springer Berlin Heidelberg.

Bencina, Ross. 2005. 'The Metasurface - Applying Natural Neighbour Interpolation to Two-to-Many Mapping'. In *Proceedings of the 2005 International Conference on New Interfaces for Musical Expression*. Vancouver.

Best, Paul, Jean Bresson, and Diemo Schwarz. 2018. 'Musical Gesture Recognition Using Machine Learning and Audio Descriptors'. In *2018 International Conference on Content-Based Multimedia Indexing (CBMI)*, 1–4. La Rochelle: IEEE. https://doi.org/10.1109/CBMI.2018.8516474.

Bigo, Louis, Antoine Spicher, and Olivier Michel. 2010. 'Spatial Programming for Music Representation and Analysis'. In *2010 Fourth IEEE International Conference on Self-Adaptive and Self-Organizing Systems Workshop*, 98–103. Budapest, TBD, Hungary: IEEE. https://doi.org/10.1109/SASOW.2010.22.

Birnbaum, David, Rebecca Fiebrink, Joseph Malloch, and Marcelo M. Wanderley. 2017. '2005: Towards a Dimension Space for Musical Devices'. In *A NIME Reader*, 3:211–22. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-47214-0_14.

Blondeau, Julia. 2017. 'Espaces Compositionnels et Temps Multiples : De La Relation Forme/Matériau'. Thesis, Paris: École doctorale Informatique, télécommunications et électronique, Paris 6. http://www.theses.fr/2017PA066292.

Boden, Margaret A. 2003. *The Creative Mind: Myths and Mechanisms*. 2nd edition. London ; New York: Routledge.

Bogost, Ian. 2008. 'The Rhetoric of Video Games'. *The Ecology of Games: Connecting Youth, Games, and Learning*, 23. https://doi.org/10.1162/dmal.9780262693646.117.

Boon, Mieke, and Tarja Knuuttila. 2009. 'Models as Epistemic Tools in the Engineering Sciences: A Pragmatic Approach.' In *Handbook of the Philosophy of Science.*, 9: Philosophy of Technology and Engineering:687–720.

Bown, Oliver, Alice Eldridge, and Jon McCormack. 2009. 'Understanding Interaction in Contemporary Digital Music: From Instruments to Behavioural Objects'. *Organised Sound* 14 (2): 188–96. https://doi.org/10.1017/S1355771809000296.

Brandtsegg, Øyvind, Sigurd Saue, and Thom Johansen. 2011. 'A Modulation Matrix for Complex Parameter Sets'. *NIME 2011 Proceedings of the International Conference on New Interfaces for Musical Expression*, no. 7491: 316–19.

Bullock, Jamie, and Ali Momeni. 2015. 'Ml.Lib: Robust, Cross-Platform, Open-Source Machine Learning for Max and PureData'. In *Proceedings of the International Conference on New Interfaces for Musical Expression*. Baton Rouge. https://pdfs.semanticscholar.org/7d55/8433925a5ec7203a6d19b754c45f4b959173.pdf.

Cage, John. 1961. *Silence: Lectures and Writings*. Middletown, Conn: Wesleyan.

Campo, Alberto de. 2014. 'Lose Control, Gain Influence - Concepts for Metacontrol'. In *Proceedings ICMC|SMC|2014*, 6. Athens.
Chaudhary, Amar, Adrian Freed, and Lawrence A Rowe. n.d. 'OpenSoundEdit: An Interactive Visualization and Editing Framework for Timbral Resources', 5.
Collins, Nicholas. 2006. 'Towards Autonomous Agents for Live Computer Music : Realtime Machine Listening and Interactive Music Systems'. *Journal of New Music Research*, no. October 2003: 245–245.
Csikszentmihalyi, Mihaly. 2008. *Flow: The Psychology of Optimal Experience*. 1 edition. New York: Harper Perennial Modern Classics.
Dagleish, Mathew. 2013. 'A Contemporary Approach to Expressivenessin the Design of Digital Musical Instruments'. Wolverhampton: University of Wolverhampton.
Dahlstedt, Palle. 2001. 'A MutaSynth in Parameter Space: Interactive Composition through Evolution'. *Organised Sound* 6 (2): 121–24. https://doi.org/10.1017/S1355771801002084.
De Landa, Manuel. 2011. *Philosophy and Simulation: The Emergence of Synthetic Reason*. London ; New York, NY: Continuum.
Dean, Roger. 2003. *Hyperimprovisation: Computer-Interactive Sound Improvisation*. Middleton: A-R Editions, Inc.
Duch, Michael. 2015. 'Free Improvisation - Method an Genre'. *Research Catalogue*, June. https://www.researchcatalogue.net/view/110382/110595.
Egido, Fernando. 2015. 'What Is Cognitive Parametric Music?' *V!RUS*, no. 11: 4.
Emerson, Gina, and Hauke Egermann. 2018. 'Exploring the Motivations for Building New Digital Musical Instruments'. *Musicae Scientiae*, October, 102986491880298. https://doi.org/10.1177/1029864918802983.
Emmerson, Simon. 1998. 'Aural Landscape: Musical Space'. *Organised Sound* 3 (2): 135–40. https://doi.org/10.1017/S1355771898002064.
Evens, Aden. 2006. 'Object-Oriented Ontology, or Programming's Creative Fold'. *Angelaki* 11 (1): 89–97. https://doi.org/10.1080/09697250600797930.
Fiebrink, Rebecca. 2017. 'Machine Learning as Meta-Instrument: Human-Machine Partnerships Shaping Expressive Instrumental Creation'. In *Musical Instruments in the 21st Century: Identities, Configurations, Practices*, edited by Till Bovermann, Alberto de Campo, Hauke Egermann, Sarah-Indriyati Hardjowirogo, and Stefan Weinzierl, 137–51. Singapore: Springer Singapore. https://doi.org/10.1007/978-981-10-2951-6_10.
Forth, Jamie, Alex Mclean, and Geraint Wiggins. 2008. 'Musical Creativity on the Conceptual Level'. In *Proceedings of International Joint Workshop on Computational Creativity 2008*. Madrid.
Forth, Jamie, Geraint A. Wiggins, and Alex McLean. 2010. 'Unifying Conceptual Spaces: Concept Formation in Musical Creative Systems'. *Minds and Machines* 20 (4): 503–32. https://doi.org/10.1007/s11023-010-9207-x.
Freed, Adrian, John MacCallum, Andrew Schmeder, and David Wessel. 2010. 'Visualizations and Interaction Strategies for Hybridization Interfaces', 5.
Gärdenfors, Peter. 2000. *Conceptual Spaces: The Geometry of Thought*. Cambridge, Mass: MIT Press.
Garnett, Guy E., and Camille Goudeseune. 1999. 'Performance Factors in Control of High-Dimensional Space'. *ICMC Proceedings 1999*.
Giavitto, Jean-Louis. 2015. 'A Topological Approach of Musical Relationships'. In *Mathemusical Conversations*. Singapour, Singapore: National University of Singapore and Institute for Mathematical Science (Singapore) and Yong Siew Toh Conservatory of Music. https://hal.archives-ouvertes.fr/hal-01257546.
Giavitto, Jean-Louis, and Olivier Michel. 2002. 'Data Structure as Topological Spaces'. In *Unconventional Models of Computation*, 2509:137–50. Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/3-540-45833-6_12.
Gillian, Nicholas. 2011. 'Gesture Recognition for Musician Computer Interaction'. Belfast: Faculty of

Arts, Humanities and Social Sciences School of Music & Sonic Arts, Queen's University Belfast.
Godøy, Rolf Inge. 2010. *Musical Gestures: Sound, Movement, and Meaning*. 1st ed. Routledge. https://doi.org/10.4324/9780203863411.
Goldmann, Stefan. 2015. *Presets – Digital Shortcuts to Sound*. London & Berlin: The Tapeworm.
Grey, John M. 1977. 'Multidimensional Perceptual Scaling of Musical Timbres'. *The Journal of the Acoustical Society of America* 61 (5): 1270–77. https://doi.org/10.1121/1.381428.
Gunnarsson, Bjarni. 2012. 'Processes and Potentials: Composing Through Objects, Networks and Interactions'. The Hague: Institute of Sonology, Royal Conservatoire.
———. 2019. 'Context and Scope'. 2019. https://www.researchcatalogue.net/view/540350/540351.
Harman, Graham. 2018. *Object-Oriented Ontology: A New Theory of Everything*. Pelican.
Harper, Adam. 2009. 'Rouge's Foam: Review: David Stubbs, "Fear of Music: Why People Get Rothko But Don't Get Stockhausen"'. *Rouge's Foam* (blog). 17 April 2009. http://rougesfoam.blogspot.com/2009/04/review-david-stubbs-fear-of-music-why.html.
———. 2011. *Infinite Music: Imagining the Next Millennium of Human Music-Making*. Winchester - Washington: Zero Books.
Hayles, N Katherine. 2016. 'Cognitive Assemblages: Technical Agency and Human Interactions'. *Critical Inquiry*, no. 43 (Autumn): 25.
Herber, Norbert F. 2010. 'Amergent Music: Behavior and Becoming in Technoetic & Media Arts'. *Media*. http://pearl.plymouth.ac.uk:8080/handle/10026.1/307.
Hildebrand Marques Lopes, Dominik, Hannes Hoelzl, and Alberto de Campo. 2017. 'Three Flavors of Post-Instrumentalities: The Musical Practices of, and a Many-Festo by Trio Brachiale'. In *Musical Instruments in the 21st Century: Identities, Configurations, Practices*, 335–60. https://doi.org/10.1007/978-981-10-2951-6_22.
Hofstadter, Douglas R. 1995. *Fluid Concepts & Creative Analogies: Computer Models of the Fundamental Mechanisms of Thought*. New York: Basic Books.
Hunt, Andrew John, and Ross Kirk. 2000. 'Mapping Strategies for Musical Performance'. In *Trends in Gestural Control of Music*, edited by Marcelo Wanderley and Marc Battier. Paris: IRCAM - Centre Pompidou.
'IMPS The Interactive Musical Prediction System'. n.d. Creative Prediction with Neural Networks. Accessed 31 May 2019. http://creativeprediction.xyz/imps/.
Jehan, Tristan. 2005. 'Creating Music by Listening'. Cambridge, MA: Massachusetts Institute of Technology.
Jordà, Sergi. 2004. 'Instruments and Players: Some Thoughts on Digital Lutherie'. *Journal of New Music Research* 33 (3): 321–41. https://doi.org/10.1080/0929821042000317886.
Kanaga, David. 2018. 'Ecooperatic Music Game Theory'. *The Oxford Handbook of Algorithmic Music*, February. https://doi.org/10.1093/oxfordhb/9780190226992.013.11.
Kreković, Gordan, and Antonio Posćić. 2015. 'Shaping Microsound Using Physical Gestures'. *Conference on Computation, Communication, Aesthetics and X*.
Macedo, Dr Frederico. 2014. 'Space as Reference: Representations of Space in Electroacoustic Music'. *JMM: The Journal of Music and Meaning, Vol. 12, 2013/2014*, 26.
Macedo, Frederico. 2015. 'Space as Metaphor: The Use of Spatial Metaphors in Music and Music Writing'. *Signata*, no. 6 (December): 215–30. https://doi.org/10.4000/signata.1085.
Magnusson, Thor. 2010. 'An Epistemic Dimension Space for Musical Devices'. In *Proceedings of the 2010 Conference on New Interfaces for Musical Expression (NIME 2010)*, 4. Sydney.
———. 2019. *Sonic Writing: Technologies of Material, Symbolic & Signal Inscriptions*. New York: Bloomsbury Academic.
Marier, Martin. 2012. 'Designing Mappings for Musical Interfaces Using Preset Interpolation'. In *Proceedings of the New Interfaces for Musical Expression (NIME) 2012 Conference*, 4. Ann

Arbor.
Martin, Charles P, and Jim Torresen. 2019. ‘Predictive Musical Interaction with MDRNNs’, 5.
Mazzola, Guerino. 2003. *The Topos of Music: Geometric Logic of Concepts, Theory, and Performance*. Basel: Birkhäuser.
Mazzola, Guerino, and Moreno Andreatta. 2007. ‘Diagrams, Gestures and Formulae in Music’. *Journal of Mathematics and Music* 1 (1): 23–46. https://doi.org/10.1080/17459730601137716.
McAdams, Stephen, and Bruno L. Giordano. 2012. *The Perception of Musical Timbre*. Edited by Susan Hallam, Ian Cross, and Michael Thaut. Vol. 1. Oxford University Press. https://doi.org/10.1093/oxfordhb/9780199298457.013.0007.
McCormack, J, Alice Eldridge, Alan Dorin, and P McIlwain. 2012. ‘Generative Algorithms for Making Music: Emergence, Evolution, and Ecosystems’. *The Oxford Handbook of Computer Music*, January. https://doi.org/10.1093/oxfordhb/9780199792030.013.0018.
McLean, Alex, and Geraint A. Wiggins. 2010. ‘Bricolage Programming in the Creative Arts’. In *PPIG*, 12.
Medeiros, Rodrigo, Filipe Calegario, Giordano Cabral, and Geber Ramalho. 2014. ‘Challenges in Designing New Interfaces for Musical Expression’. In *Design, User Experience, and Usability. Theories, Methods, and Tools for Designing the User Experience*, edited by Aaron Marcus, 8517:643–52. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-07668-3_62.
Miranda, Eduardo Reck, ed. 2011. *A-Life for Music: Music and Computer Models of Living Systems*. The Computer Music and Digital Audio Series 24. Middleton, Wis: A-R Editions.
Momeni, Ali, and Cyrille Henry. 2006. ‘Dynamic Independent Mapping Layers for Concurrent Control of Audio and Video Synthesis’. *Computer Music Journal* 30 (1): 49–66.
Momeni, Ali, and David Wessel. 2003. ‘Characterizing and Controlling Musical Material Intuitively with Geometric Models’. In *Proceedings of the 2003 Conference on New Interfaces for Musical Expression (NIME-03), Montreal, Canada*, 54–62. Montreal.
Morton, Timothy. 2013. *Hyperobjects: Philosophy and Ecology after the End of the World*. Posthumanities 27. Minneapolis: University of Minnesota Press.
Mudd, Tom, Paul Mulholland, Simon Holland, and Nick Dalton. 2014. ‘Dynamical Interactions with Electronic Instruments’. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, 4.
Mulder, Axel G. E., S. Sidney Fels, and Kenji Mase. 1997. ‘Empty-Handed Gesture Analysis in Max/FTS’. In *In Proceedings of Kansei - The Technology of Emotion, AIMI International Workshop*, 87–91.
Mylov, Peer. 2002. ‘On Space, Its Time, and Spatiotemporal Expressions’. In , 47–70. https://doi.org/10.1007/978-1-4471-0225-0_3.
Naveda, Luiz, and Marc Leman. 2010. ‘The Spatiotemporal Representation of Dance and Music Gestures Using Topological Gesture Analysis (TGA)’. *Music Perception* 28 (1): 93–111. https://doi.org/10.1525/mp.2010.28.1.93.
Oord, Aaron van den, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. 2016. ‘WaveNet: A Generative Model for Raw Audio’. *ArXiv:1609.03499 [Cs]*, September. http://arxiv.org/abs/1609.03499.
Oppenheim, Daniel. 1995. ‘“DMorph”: An Interactive System for Compositional Morphing of Music in Real-Time’. Unpublished. https://doi.org/10.13140/rg.2.1.2494.5689.
O’Sullivan, Liam. 2013. ‘MorphOSC- A Toolkit for Building Sound Control GUIs with Preset Interpolation’. In , 6.
Paine, Garth. 2009. ‘Towards Unified Design Guidelines for New Interfaces for Musical Expression’. *Organised Sound* 14 (2): 142–55. https://doi.org/10.1017/S1355771809000259.
Pearce, Marcus, David Meredith, and Geraint Wiggins. 2002. ‘Motivations and Methodologies for Automation of the Compositional Process’. *Musicae Scientiae* 6 (2): 119–47. https://doi.org/

10.1177/102986490200600203.
Pei, Ling, Jingbin Liu, Robert Guinness, Yuwei Chen, Heidi Kuusniemi, and Ruizhi Chen. 2012. 'Using LS-SVM Based Motion Recognition for Smartphone Indoor Wireless Positioning'. *Sensors* 12 (5): 6155–75. https://doi.org/10.3390/s120506155.
Pendharkar, Chinmay, Michael Gurevich, and Lonce Wyse. 2013. 'Parameterized Morphing as a Mapping Technique for Sound Synthesis'. In *Proceedings of the 9th International Conference on Digital Audio Effects, DAFx 2006*.
Poupyrev, Ivan, Michael J. Lyons, Sidney Fels, and Tina Blaine (Bean). 2001. 'New Interfaces for Musical Expression'. In *CHI '01 Extended Abstracts on Human Factors in Computer Systems - CHI '01*, 491. Seattle, Washington: ACM Press. https://doi.org/10.1145/634342.634348.
Priestley, John. 2014. 'Poiesthetic Play in Generative Music'. Virginia Commonwealth University.
Rheinberger, Hans-Jörg. 2015. 'Experimental Systems: Difference, Graphematicity, Conjuncture'. 2015. https://dirnagl.files.wordpress.com/2015/04/rheinberger_experimental_systems_engl.pdf. Accessed 10 May 2019.
Roads, Curtis. 2015. *Composing Electronic Music*. 1 edition. Oxford ; New York: Oxford University Press.
Roberts, Adam, Jesse Engel, and Douglas Eck. 2017. 'Hierarchical Variational Autoencoders for Music'. In *Machine Learning for Creativity and Design - NIPS 2017 Workshop*, 256:1–6.
Roberts, Adam, Jesse Engel, Sageev Oore, and Douglas Eck, eds. 2018. 'Learning Latent Representations of Music to Generate Interactive Musical Palettes'. In *Joint Proceedings of the ACM IUI 2018 Workshops*. Tokyo. http://ceur-ws.org/Vol-2068/milc7.pdf.
Roberts, Adam, Jesse Engel, Colin Raffel, Curtis Hawthorne, and Douglas Eck. 2018. 'A Hierarchical Latent Vector Model for Learning Long-Term Structure in Music'. *ArXiv:1803.05428 [Cs, Eess, Stat]*, March. http://arxiv.org/abs/1803.05428.
Rowe, Robert. 1992. *Interactive Music Systems: Machine Listening and Composing*. Cambridge, Mass: MIT Press.
Rutz, Hanns Holger. 2016a. 'Marking a Space of Algorithmicity'. In *Proceedings of the 4th Conference on Computation, Communication, Aesthetics & X (XCoAx)*, 14. Bergamo.
———. 2016b. 'Agency and Algorithms'. *Journal of Science and Technology of the Arts* 8 (1): 73. https://doi.org/10.7559/citarj.v8i1.223.
Ryan, Joel. 2002. 'Musical Visualization. Prospecting for New Performance and Composition Tools for Electronic Music'. https://jr.home.xs4all.nl/MuViz.htm.
Sanfilippo, Dario. 2017. 'Environment-Mediated Coupling of Autonomous Sound-Generating Systems in Live Performance: An Overview of the Machine Milieu Project.' In *Proceedings of the 14th Sound and Music Computing Conference*, 7. Espoo.
Schaeffer, Pierre. 2017. *Treatise on Musical Objects: An Essay across Disciplines*. First edition. Oakland, California: University of California Press.
Schnell, Norbert, and Marc Battier. 2002. 'Introducing Composed Instruments, Technical and Musicological Implications'. In *Proceedings of the 2002 Conference on New Interfaces for Musical Expression*, 1–5. NIME '02. Singapore, Singapore: National University of Singapore. http://dl.acm.org/citation.cfm?id=1085171.1085205.
Schwab, Michael, Virginia Anderson, Paulo de Assis, Elke Bippus, Henk Borgdorff, Darla Crispin, Paolo Giudici, et al., eds. 2013. *Experimental Systems: Future Knowledge in Artistic Research*. Orpheus Institute Series. Leuven: Leuven University Press.
Schwartz, Elliott, Barney Childs, and Jim Fox. 1998. *Contemporary Composers On Contemporary Music*. Expanded edition. New York: Da Capo Press.
Schwarz, Diemo. 2012. 'Navigating Variation: Composing for Audio Mosaicing', 4.
Seago, A. 2008. 'TIMBRE SPACE AS SYNTHESIS SPACE: TOWARDS A NAVIGATION BASED APPROACH TO TIMBRE SPECIFICATION'. *Proceedings of the Institute of Acoustics* 30: 8.

Shao, Jianhua, James Mcdermott, Michael O'Neill, and Anthony Brabazon. 2010. 'Jive: A Generative, Interactive, Virtual, Evolutionary Music System'. In , 6025:341–50. https://doi.org/10.1007/978-3-642-12242-2_35.
Smalley, Denis. 1994. 'Defining Timbre — Refining Timbre'. *Contemporary Music Review* 10 (2): 35–48. https://doi.org/10.1080/07494469400640281.
———. 1997. 'Spectromorphology: Explaining Sound-Shapes'. *Organised Sound* 2 (2): 107–26. https://doi.org/10.1017/S1355771897009059.
———. 2007. 'Space-Form and the Acousmatic Image'. *Organised Sound* 12 (1): 35–58. https://doi.org/10.1017/S1355771807001665.
Stewart, Ian. 1976. 'Catastrophe Theory'. 1976. http://www.thebookshelf.auckland.ac.nz/docs/Maths/PDF/mathschron005-013.pdf.
Stoiber, Nicolas, Renaud Seguier, and Gaspard Breton. 2008. 'Automatic Design of a Control Interface for a Synthetic Face'. In *Proceedingsc of the 13th International Conference on Intelligent User Interfaces - IUI '09*, 207. Sanibel Island, Florida, USA: ACM Press. https://doi.org/10.1145/1502650.1502681.
Tahiroğlu, M. Koray. 2008. 'Interactive Performance Systems: Experimenting with Human Musical Interaction'. Helsinki: University of Art and Design Helsinki.

Tatar, Kıvanç, and Philippe Pasquier. 2019. 'Musical Agents: A Typology and State of the Art towards Musical Metacreation'. *Journal of New Music Research* 48 (1): 56–105. https://doi.org/10.1080/09298215.2018.1511736.
Teruggi, Daniel. 2007. 'Technology and Musique Concrète: The Technical Developments of the Groupe de Recherches Musicales and Their Implication in Musical Composition'. *Organised Sound* 12 (3): 213–31. https://doi.org/10.1017/S1355771807001914.
Thio, Vibert, Hao-Min Liu, Yin-Cheng Yeh, and Yi-Hsuan Yang. 2019. 'A Minimal Template for Interactive Web-Based Demonstrations of Musical Machine Learning'. *ArXiv:1902.03722 [Cs]*, February. http://arxiv.org/abs/1902.03722.
Toiviainen, Petri, Mauri Kaipainen, and Jukka Louhivuori. 1995. 'Musical Timbre: Similarity Ratings Correlate with Computational Feature Space Distances*'. *Journal of New Music Research* 24 (3): 282–98. https://doi.org/10.1080/09298219508570686.
Tubb, Robert, and Simon Dixon. 2014. 'A Zoomable Mapping of a Musical Parameter Space Using Hilbert Curves'. *Computer Music Journal* 38 (3): 23–33. https://doi.org/10.1162/COMJ_a_00254.
Tubb, Robert H. 2010. 'Creativity, Exploration and Control in Musical Parameter Spaces'. London: School of Electronic Engineering and Computer Science, Queen Mary University of London.
Tymoczko, Dmitri. 2011. *A Geometry of Music: Harmony and Counterpoint in the Extended Common Practice*. Oxford Studies in Music Theory. New York: Oxford University Press.
Ungeheuer, Elena. 1994. 'From the Elements to the Continuum: Timbre Composition in Early Electronic Music'. *Contemporary Music Review* 10 (2): 25–33. https://doi.org/10.1080/07494469400640271.
Vertegaal, Roel, and Ernst Bonis. n.d. 'ISEE: An Intuitive Sound Editing Environment'. Accessed 31 May 2019. https://artsdocbox.com/Music/65767888-Isee-an-intuitive-sound-editing-environment.html.
'Wave Mapper Manual.Pdf'. n.d. Accessed 31 May 2019. http://wolfgangpalm.com/mediafiles/2013/01/WaveMapper_Manual.pdf.
Wessel, David. 1991. 'Improvisation with Highly Interactive Real-Time Performance Systems'. In *Proceedings of the International Computer Music Conference*, 344–344. INTERNATIONAL COMPUTER MUSIC ACCOCIATION.
Wessel, David L. 1979. 'Timbre Space as a Musical Control Structure'. *Computer Music Journal* 3 (2): 45. https://doi.org/10.2307/3680283.

Wiggins, Geraint A. 2018a. 'Creativity, Information, and Consciousness: The Information Dynamics of Thinking'. *Physics of Life Reviews*. https://doi.org/10.1016/j.plrev.2018.05.001.

———. 2018b. 'Creativity, Information, and Consciousness: The Information Dynamics of Thinking'. *Physics of Life Reviews*, May. https://doi.org/10.1016/j.plrev.2018.05.001.

Wiggins, Geraint A., and Jamie Forth. 2018. *Computational Creativity and Live Algorithms*. Edited by Roger T. Dean and Alex McLean. Vol. 1. Oxford University Press. https://doi.org/10.1093/oxfordhb/9780190226992.013.19.

Wilson, Peter Niklas. 2014. *Hear and Now: Gedanken zur improvisierten Musik*. Hofheim: Wolke V.-G.

Wilson, Scott, Norah Lorway, Rosalyn Coull, Konstantinos Vasilakos, and Tim Moyers. 2014. 'Free as in BEER: Some Explorations into Structured Improvisation Using Networked Live-Coding Systems'. *Computer Music Journal* 38 (1): 54–64. https://doi.org/10.1162/COMJ_a_00229.

Wishart, Trevor. 1996. *On Sonic Art*. New and rev. ed. Contemporary Music Studies, v. 12. Amsterdam: Harwood Academic Publishers.

Wooller, René, and Andrew R Brown. 2005. 'Investigating Morphing Algorithms for Generative Music'. *Procedings of Third Iteration*, 10.

Xenakis, Iannis. 1992. *Formalized Music. Thought and Mathematics in Composition*. 2nd edition. Harmonologia Series (Book 6). Stuyvesant, NY: Pendragon Press.

Zbikowski, Lawrence Michael. 2002. *Conceptualizing Music: Cognitive Structure, Theory, and Analysis*. AMS Studies in Music. Oxford ; New York: Oxford University Press.

Appendix A: Technical Implementation

The core of the system is programmed in *SuperCollider,* an open-source object-oriented language and environment for algorithmic music and sound programming. In *SuperCollider* everything is an object, consisting of data and methods used to operate with them. Objects can be functions, signal processing unit generators, data structures, graphical elements, wave samples, etc.[50] *SuperCollider* has an extensive collection of the so-called pattern classes, powerful high-level algorithms aimed at generative and stochastic sequencing of events. By virtue of the JIT Lib (Just In Time) library it is furthermore possible to modify the processes but also to swap whole pattern definitions or their parts on-the-fly during the performance. This can be an important feature contributing to the open-ended character and real-time capabilities of the system. The open-source nature and community-based development of *SuperCollider* also enabled me to take advantage of several available libraries and third-party classes such as the *PresetInterpolator* (creation of the interpolation *control space*), *VSTPlugin* (hosting Virtual Software Instrument synthesizers), *Modality Toolkit* (efficient management of input interfaces), or *OnlineMIDI* class that I adapted to be used with the OSC protocol (real-time listetning and analysis of event data streams).

For illustration I enclose an example of Behaviour definition in the *SuperCollider's* sclang language together with patterns used for event and envelope generation:

```
~behaviour[1] = (

    patterns: (
            eventpats: ~currentArraypats[1],
            envelopes: ~currentEnvelopes[1],
        ),
        synthparams: [0.22, 0.1123, 0.421, …, 0.4311], // array containing 86 synthesis parameter
settings
    global: (
            loudness
            speed: 0.4,
            density: 0.1,
```

[50] http://doc.sccode.org/Guides/Intro-to-Objects.html

```
                entropy: 0.0,
                pitchdeviation: 0.5,
                envdeviation: 0.3,
                notes: 60,
                transpose: 0,
                loudness: 0.7
            )
);
~currentArraypats[1] = Pdef(\p7,
    Pbind(
        \midinote, ~gP.notesP + ~gP.pitchdeviationP,
        \dur, ~lsys3 * ~gP.speedP + ~gP.entropyP,
        \legato, Pwhite(0.4, 0.8, inf) + ~gP.entropyP,
        \amp, Pwrand([-1.0, Pwhite(0.7, 1.0, 1)], Pfunc({([(1-~g.density).clip(0.0, 1.0),
            ~g.density].normalizeSum * [0.1, 0.9]).normalizeSum}), inf) * ~g.loudness
));
~envelopes[1] = Ptuple([
        Prand([50, 65, 70], inf) + Pwhite(0.0, ~gP.envdeviationP * -30.0, inf), // attack
            Prand([40, 60, 80], inf), // decay
            Prand([40], inf), // sustain
            Pn(Pseq([ Pseries(30, 1, 90), Pseries(120, -1, 90)]), inf) +
                Pwhite(0.0, ~gP.envdeviationP * 100.0, inf)]) // release
);
```

To be able to dynamically morph between several event generating patterns based on the **Cursor** position, I implemented a pattern interpolation function taking an array of patterns and current weights for the respective **Locations** as arguments:

```
~morphpat =
;
    morphroutine = Routine {
        block { arg break;
            loop {
                ev = ();
                streampat.size.do{|i| ev[i] = streampat[i].next(Event.default) };
                result = ();
                ev[0].keys.difference(exclu).do
.sum(ev.size)
                };
               result.yield;
            }
        }
    };
    morphroutine
};
```

Parameter Mapping

The dimensionality reduction between the high-dimensional synthesis parameter space (ranging up to 100 dimensions) and the three-dimensional control space is realised using the Intersecting N-Spheres interpolation method that is available through the *SuperCollider*'s *Interpolator* and *PresetInterpolator* classes programmed by Martin Marier. The classes also provide a graphical user interface representing an interpolation surface where the points can be positioned. In his paper describing the method Marier (Marier 2012) gives an overview of previously available interpolation techniques and tools for preset morphing and poses several desired features of an interpolation method which were not always met:

> *The interpolated surface is continuous. […]*
>
> *The interpolated surface is continuously differentiable. The surface is smooth; there are no singularities. […]*
>
> *The system is autonomous. The only variables are the data points. […]*
>
> *The data points can be positioned freely. A user can put any number of data points on the surface, and he can position them anywhere. […]*
>
> *Interpolation is local. Only a limited number of points (the nearest ones) are used for the interpolation. […]*
>
> *The interpolated surface goes through the data points. The value of the interpolation on a data point is equal to the value of that data point. […]*

His method has been tested for intepolation in spaces of up to 10 dimensions. It is computationally efficient for and fulfills the aforementioned requirements which makes it very suitable for my proposed system design. Furthermore, it enables a simultaneous editing and navigating the space without the sound being interrupted, which also allows for dynamic changes of the space's geometry.

Besides the mapping of X,Y and pressure coordinates of the surface sensor input to the position of the **Cursor** in the *control space*, other mappings are realised as combinations of simple one-to-one correspondences (such as pressure/force to amplitude) and higher level input features (speed of the gesture) mapped to global parameters of the processes (tempo, envelope deviation setting). The foot pedal input is mapped dynamically and can be switched from the manipulation of morphing between the recalled **States** to the control of larger-scale **Cursor** movement on the Z-axis in the *control space*.

Connectivity

Although the Sensel Morph sensor interfaces can send MIDI messages with positions and pressure values of the contact points that could be captured and used in SuperCollider, much more detailed and higher-resolution information for can be obtained by accessing the devices directly through their API (Application Programming Interface). Based on this data computed already in the device itself (positions of the contact points with their individual pressure, covered surface area for a particular point, time deltas for each direction of movement, as well as accelerometer data), more high-level features can be extracted (maximum distance of the points, total surface area defined by all the touches, length of a convex hull that includes all the points, pitch, roll and yaw data from the accelerometer). For this purpose I programmed a utility in OpenFrameworks/C++ that sends the feature data to the system core as OSC messages at a selectable frame rate (30 frames per second work fine). The foot pedal and the grid-based controller (Launchpad Pro) both deliver MIDI data. Furthermore, sythesis and sequencing parameter values can be sent from the system core over MIDI or OSC protocols to various software or hardware synthesizers, yielding different timbral options.

Artificial Neural Network Agent

The autonomous agent generating gestures mapped to the Cursor in the *control space* is realised with an external artificial neural network model programmed by Charles Martin in the Keras and Tensorflow framework[51]s for deep learning (Martin and Torresen 2019). It is a Mixture-Density Recurrent Neural Network (MDRNN), a novel typef of network structure capable of learning and predicting gestures in multi-dimensional spaces.[52] Python scripts for the interaction with the model are provided with the the IMPS (Interactive Music Prediction System) code package,[53] enabling a bi-directional communication with a gestural digital interface of choice via OSC protocol. I adapted the scripts of the prediction server to be able to interface with the network in real-time by adding the

[51] https://github.com/cpmpercussion/keras-mdn-layer
[52] More information about the network is in the paper (Martin and Torresen 2019).
[53] https://github.com/cpmpercussion/imps

remote control option for ”temperature” settings and interaction mode switching (“call and response” mode when the network starts making gestures after the user stops, ”battle” mode meaning simultaneous data input and output, or ”user-only” mode used for logging the user input data that can be used for training). These options can now be manipulated via OSC messages sent from *SuperCollider*. Two “temperature“ parameters, named as sigma and pi temperatures define the options for sampling from the probability distribution during prediction making. Essentially they influence the degree of randomness in the output gestures. Much fine-tuning and adjustment can be done in respect to the desired results. Figure 37 shows the scheme of interaction with the network model.

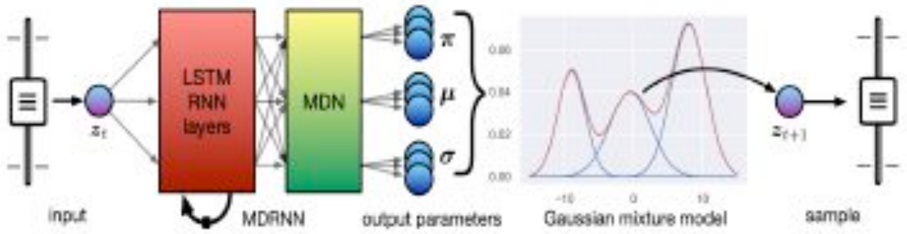

Fig. 41. Interaction with a Mixture Density Neural Network Model
(‘IMPS The Interactive Musical Prediction System’).

I chose a model of a small size and trained it on my performance gesture based on 30 minutes of playing with the AMEN system (around 48 000 samples). The training set comprised data from the gestural movement in 3D space together with finger pressure and time, so it was five dimensions in total. The training took several hours to complete on a Macbook Pro Retina 2015 model with 2.8 GHz Intel Core i7 processor and 16 GB RAM using CPU for the computations.

Sound generation

As of present, various VST (Virtual Software Instrument) plugins (Waldorf Largo, Madrona Labs Aalto and others) and *SuperCollider's* native unite generators have been tested for the sound generation. Since the core engine is independent from synthesis, it is nevertheless possible to connect external MIDI synthesizers and devices accepting control-voltage input through a digital to analog converter. In the current setup the system is used to control up to 86 synthesis parameters simultaneously, although more are possible.

Appendix B: List of Peformances

This list includes performances that took place during my studies between September 2017 and May 2019.

2017

Sonology Electroacoustic Ensemble @ Studio Loos, The Hague, 28.10.

Solo @ CASS Concert, The Royal Conservatory, The Hague, 7.11.

Sonology Electroacoustic Ensemble @ The Royal Conservatory, The Hague, 6.12.

2018

w/ Bara Latalova (dance) @ Art's Birthday 2018, Provoz Hlubina, Ostrava + EBU Radio broadcast, 17.1.

Sonology Electroacoustic Ensemble @ The Royal Conservatory, The Hague, 21.2.

Solo @ FUGA, Bratislava, 15.4.

Solo @ UrbsArt Concreta Festapoesia 2018, Accademia d'Ungheria, Rome, 21.4.

Solo @ Rear Ear / STEIM, Amsterdam, 21.5.

Solo @ Nová synagóga, Žilina, 24.5.

Mega-Phone Ensemble @ Fanfara Hranice, Kravin Rural Arts Festival, Hranice, 26.5.

Vritti @ A4, Bratislava, 8.9.

Succour Soundtracks for a soothing dystopia (2059-64) w/ Gívan Belá, Geza Bobb, Tetsuo Kogawa, Marco Kuhn, Giulia Fournier-Mercadante, Camilla Milena Fehér, Jasmina Al-Qaisi @ Dystopie Sound Art Festival, Berlin, 22.9.

Duo w/ Görkem Arıkan @ Wavelength Festival, Leiden, 6.10.

Duo w/ Görkem Arıkan @ Royal Conservatory, Den Haag, 17.10.

Shibuya Motors w/ Didi Kern & Balazs Pandi @ Fuga, Bratislava, 1.11.

Shibuya Motors w/ Didi Kern & Balazs Pandi @ Fluc, Vienna, 2.11.

Shibuya Motors w/ Didi Kern & Balazs Pandi @ Alternativa Festival, Prague, 3.11.

2019

w/ Görkem Arıkan @ Villa K, The Hague, 23.1.

Vritti @ Urbsounds Label Night, Fuga, Bratislava, 7.5.