Circle Squared and Circle Keys: Performing on and with an unstable live algorithm for the Disklavier

Palle Dahlstedt
Dept. of Applied IT & The Academy of Music and Drama
University of Gothenburg, Sweden
palle.dahlstedt@gu.se

ABSTRACT

Two related versions of an unstable live algorithm for the Disklavier player piano are presented. The underlying generative feedback system consists of four virtual musicians, listening to each other in a circular configuration. There is no temporal form, and all parameters of the system are controlled by the performer through an intricate but direct mapping, in an attempt to combine the experienced musician's physical control of gesture and phrasing, with the structural complexities and richness of generative music. In the first version, Circle Squared, the interface is an array of pressure sensors, and the performer performs on the system without participating directly, like a puppet master. In the second version, control parameters are derived directly from playing on the same piano that performs the output of the system. Here, the performer both plays with and on the system in an intricate dance with the unpredictable output of the unstable virtual ensemble. The underlying mapping strategies are presented, together with the structure of the generative system. Experiences from a series of performances are discussed, primarily from the perspective of the improvising musician.

Keywords

mapping, phrasing, gestural control, generative music, multi-agent system, feedback, performance, improvisation

1. INTRODUCTION

In the last few decades, extensive effort has been spent on the development of advanced synthesis techniques and elaborate generative algorithms. But as a musician, you also want to play your instrument. This implies physical effort-based interaction, aiming for direct control of musical expression. But mapping gesture to synthesis and generative performance is a challenge. I have previously worked extensively with the development of novel mapping approaches for synthesis (e.g., [2, 4]). Here, I explore how such mappings can be applied to generative processes, combining the structural complexity of generative music with the intuitive physical control of the trained musician, resulting in organic phrasing and direct interaction with and performance on intricate note textures.

The system consists of two parts: an underlying agent system of four virtual musicians, which is inherently unstable thanks to extensive internal feedback, and a mapping engine that in addition to a dynamic parameter vectorization also imitates the effort-based expression that is the basis of most acoustic instruments, affecting dynamics, intensity and complexity. Playing such a system is a special situation, and requires very fast reactions and a good ear. In performance, I have developed strategies for exploring the state space. Without direct control and with insufficient predictive capabilities, I have to rely on trial and error, or rather on performance patterns such as probe and react, find and rest, explore and contemplate (what you have before you lose it). Or ponder and vary, discover and exploit, or possibly even: go to the limit and jump ship.

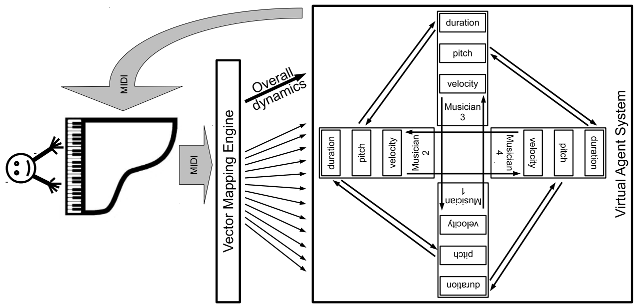

Figure 1: The overall structure of the Circle Keys system. In Circle Squared, MIDI input comes from a pressure pad controller (QuNeo or TriggerFinger), but the rest is similar.

The system exists in two versions. *Circle Squared* (CS, 2011) is controlled through 16 pressure sensors (MacMillen QuNeo), presented at Disklavier concerts in Stockholm and Gothenburg. *Circle Keys* (CK, 2013) is controlled from playing the actual piano keyboard, performed in Gothenburg, Singapore, London and Tällberg. Since the mappings and the underlying agent systems are essentially similar, they provide an opportunity to compare how the different interfaces affect the musical output. This is done through analysis of the emerging performance techniques and musical results from recordings and performances.

1.1 Previous and related works

Yamaha Disklavier pianos are equipped with MIDI in/out and electro-mechanical activators for each key. In effect, they are modern player pianos, and have inspired a number of composers. Such a piano can form a real-time link from virtual algorithms to physical reality in a way that electronic synthesizers and digital pianos cannot. There are many possible approaches to such an instrument. Conlon Nancarrow (1912-1997) sequenced complex music impossible to play and notate, in his many pieces for player piano.

NIME'14, June 30 – July 03, 2014, Goldsmiths, University of London, UK.

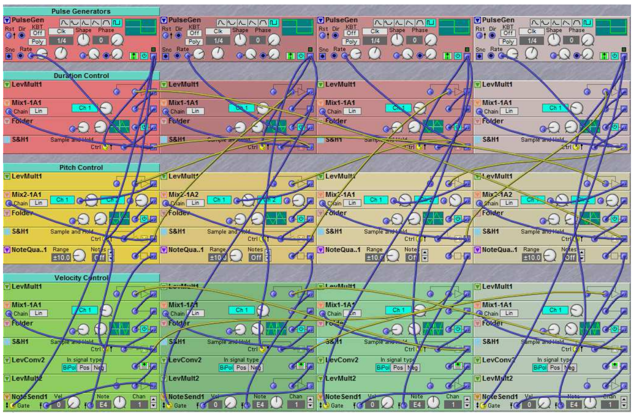

Figure 2: The system of four virtual musicians (one per column), implemented in the Nord Modular G2. Feedback connections are shown in yellow. The mapping engine is not shown.

Compositions for Disklavier can be based on real-time interactive systems, e.g., Max patches [10], statistical models [13], or complex systems that interact with the player, like the system presented in this paper. Others have based event streams on audio analysis from the piano sounds, in a feedback loop [11]. Disklaviers can also be used to perform full pieces generated from non-real-time algorithmic systems, thus bypassing the notation phase, enabling a very fast turnaround (as in [3]).

Systems of interacting agents have been used in many different contexts to producing musical structures. Low-level neuronal nets were used by Laine to produce drum patterns [8], and he has also discussed feedback in musical systems on a more general level [7]. An example of a feedback system generating musical structures, in the context if interactive evolution of system parameters, was presented in [1]. Multi-agent systems for the Disklavier, without musician interaction, have been also been presented [5].

In the presented system, the interaction between musician and agent system consists of parameter control for the feedback connections within the system, as derived from a complex dynamic mapping from interface to system parameters. The actual music played by the pianist (in CK) is not used by the system, but is still an integral part of the musical output. The musical domain of the system is atonal free improvisation, and more important than the harmony and tonality are rhythm, phrasing, gesture and structural aspects of the music.

2. IMPLEMENTATION

Both the agent system and the mapping engine are implemented in the Nord Modular G2 environment, to avoid regular computers and screens on stage, and to make the system robust, reliable, and easy to setup. The overall schematic of the setup is shown in Fig. 1.

The G2 is a DSP hardware platform (with or without keyboard and control surface), programmed through hi-resolution (24bit/96kHz) signal connections between sets of predefined modules, edited in a graphic environment. Finished patches are stored in flash memory in the DSP hardware, and can be run as stand-alone. The system is capable of both audio and MIDI synthesis and processing.

2.1 Four virtual musicians

The underlying agent system consists of four simple interconnected virtual agents, each with three internal states: current pitch, duration (expressed as a frequency of subsequent notes) and amplitude (velocity), and it is controlled through a set of interaction coefficients and some parameters that directly affect the playback (articulation, overall dynamics, etc.). The feedback connections between agent parameters are shown in Fig. 1. The feedback connections have a nonlinear component, implemented as a folding function with parameter control of the amount of folding. All agents are equal, and all interconnections follow a rotational symmetry, hence the titles *"Circle ..."*.

The feedback system design was based on experiences from autonomous generative systems designed for interactive evolutionary exploration of parameter sets [1]. This research showed that, in spite of the high potential for chaotic behavior, the parameter space for such systems could be explored by ear. Alternative connection patterns and formulas have been tried, but with less interesting results.

Each virtual musician has three parts (see Fig. 2), controlling duration, pitch and velocity. Internal states are updated based on a pulse generator that also controls duration, through rate modulation. Feedback connections are of the form $x_t = Fold(x_{t-1} + ay_{t-1}, b)$ where y is an internal state from another agent, a is the connection strength, and b is the amount of nonlinear folding, the latter two controlled from the vector mapping engine. Duration is affected by the pitch of the left neighbor. Pitch is affected by the duration of the right neighbor, and the velocity is affected by the velocity of the neighbor opposite in the circle. Articulation (note on/off proportion) is controlled directly by the mapping engine.

2.2 Mapping

As mentioned, the dynamic mapping engine is based on previous research, but this is the first time it is used to control a generative system, shaping the structure of the music. The current implementation is briefly described below.

The mapping translates from a set of control parameters (pressures or velocities) from the interface (pressure pads or MIDI keyboard) to a set of system parameters of the generative agent system. The system parameters are: a number of feedback strength coefficients between the internal states of the different agents in the agent system, the amount of non-linearity (folding) of these connections, and some parameters for articulation and dynamics.

Each control parameter controls the magnitude of its corresponding vector in system parameter space (Fig. 3a). If several pads or keys are pressed, the resulting vectors are added, resulting in a single point in parameter space, i.e., a set of system parameters (Fig. 3b). For example, by applying varying pressure to two pads, complex 2D trajectories in the system parameter space can be created. There is no limit to the number of simultaneous pads or keys played. Vectors are also scaled by a global parameter, to allow for wild exploration or minute nuances. The origin of the vector sum can be shifted to the current point at any moment, by the press of the 16th pad in the pad version or a press of a pedal in the keyboard version. It is reset by a hard/long press on the same pad/pedal.

The mapping is a many-to-many mapping, or rather, an *all-to-all* mapping, since every single change in control parameters affect all system parameters. This is based on results showing that coupled mappings are more expressive and interesting to play [6]. Further mapping research influential for my chosen approach can be found in [9, 12]. The concept of *intimate* control of the sonic output, as introduced by David Wessel et al [14], has also been important.

The mapping is dynamic, in the sense that this the player can change it on the fly. The primary ways to change it are

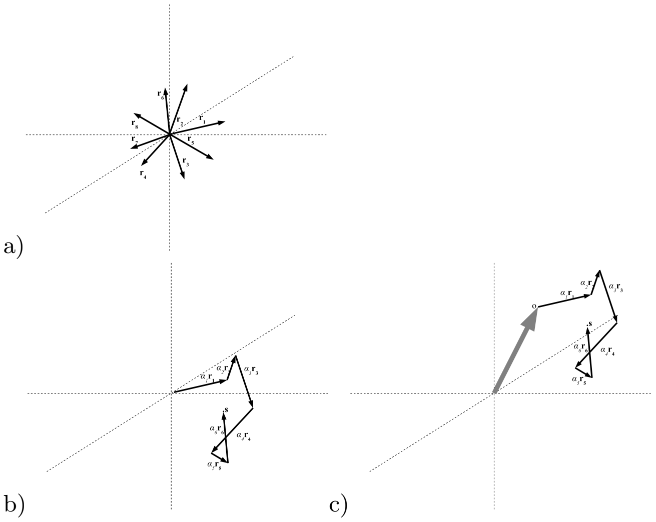

Figure 3: An illustration of the vector mapping engine (from [2]). a) A pad/key corresponds to a vector in system parameter space. Vectors are scaled by velocity/pressure. b) Vectors from simultaneous pad/key presses are summed to reach the target point. c) The vector system can be shifted, to explore different regions.

to change the vector scaling and to shift the vector origin to a new position. In this way, the player can, for example, zoom in on interesting behavior.

The mapping is *non-designed*, meaning that the specific vectors used to translate from control parameters to system parameters are not hand-tuned in any way. Instead, they are randomized at design time, with the possibility to re-randomize them at will (although done rarely). There are several reasons for this choice. First, the mapping algorithm is developed for exploratory improvisation, and the unpredictable aspects of the system provides input to the performer, which can be elaborated upon and trigger new ideas. This is especially useful in a solo context, without input from co-players. The vector set resembles a full modulation matrix, but is controlled and conceptualized differently. To set the vectors by hand would would be tedious and difficult. Approaches based on principal component analysis of actual playing have been considered, but goes against the idea of improvisation as exploration of unknown possibilities. The goal is not to repeat previous deeds but to explore and learn while playing.

3. PERFORMANCE ANALYSIS

Over a number of sessions, performances and recordings, characteristic playing techniques have emerged – from what is possible to do, what is easy to do, and the challenges that are potentially possible to overcome, in interaction with the aesthetic preferences and physical constraints of the performer. Since the underlying generative system is chaotic and unstable, the player has to be very alert at every moment. You can never rely on what will come out. Demos of performances on CS[1], and CK[2] are posted on YouTube, referenced in the following discussion.

An adequate metaphor is driving a car with control only over average speed and rate of turning, plus an emergency brake. You can always stop, but have to be careful when you start, unless you are brave or in a safe area. If it goes well, you keep going. If you see catastrophe approaching, you can turn, slow down, or stop. If you see a pattern in the movement, you can exploit this insight and dance around in more controlled patterns. Although a bit risky, you can still go somewhere with this kind of vehicle.

Although complex, the Circle system has some properties based on physical analogies, making it easier to play. E.g., pressure/velocity is mapped to volume, based on the idea that physical effort corresponds to loudness; true in most acoustic instruments. Pressures/velocities also scales vectors, with more effort resulting in larger parameter changes (from status quo). Since inter-agent connection strengths are scaled by vector parameters, more musical change follows from more effort. Vectors also controls folding amount. Hence, more effort leads to more complexity.

A specific combination of control parameters (pressures/velocities) generates a specific set of system parameters. Each pad or key affects all parameters. The mapping is not random, but deterministic, and a scenario can be repeated, with regards to system parameters. However, because of the internal states, the exact musical behavior is unpredictable. The skill acquired by extensive practice consists of learning how to quickly react to what is actually happening, and quickly detect the effects of current pressure or key combinations by cognitive extrapolation.

A key technique is to shift the vector origin when an interesting behavior is encountered, to further explore the region around this point. This can be seen in CS1, 0'44" (upper right pad press). It can clearly be heard that the following music is colored by this. The origin is reset at 0'57".

In CS, organic control of phrasing is easy. You can play very softly, then switch to very loud by increasing the pressure or hitting a pad very hard. It is also easy to play single chords or bursts of notes by distinct pad playing. Also common is to quickly shift between different textures (pad combinations) to shape a musical phrase, in something that feels like real-time composition. Reacting to what happens, you shift when new material is needed. Overall intensity is affected by pressure, and the internal states of the virtual musicians ensure a smooth transition (CS2, 0'00"-0'20").

Behavior can be frozen by holding fingers still (CS1, 2'03"), or gradually changed by slowly changing the pressure pattern. For a radical change without losing the current texture quality, a burst of change can be induced by a hard but short pad hit, while keeping others still. In this way, internal states are perturbed, but system dynamics retained (CS1, 2'09"; CK1, 0'24"-0'44").

By strong pressure or very loud chord playing, the agent system can be brought into very fast movement, rushing quickly, possibly exponentially, to the edge of the pitch register and very fast sequences. Just before total chaos, you release keys or pads, or switch to another pad or key combination. I call this to the limit, then jump ship. Repeated exploration of such patterns are possible (CS2, 4'29"-5'10"; CK2, 3'20"; CS1, 1'50").

In CK, playing any key gates the system activity, i.e., the virtual musicians freeze then nothing is played. This can be exploited by pausing a texture, bringing out just pointillistic fragments of it, or by repeated playing of short bursts of the same, or similar key combinations.

CK also allows for more controlled attacks, because the actual pressed keys produce a sound, and because of the velocity-based mapping. In CS, the focus is on kneading structures, physically and metaphorically, since vectors are scaled continuously. In CK, change happens in steps, which significantly affects the character of the musical output.

[1]CS1: `http://www.youtube.com/watch?v=6gcCLfmLyeE`, CS2: `http://www.youtube.com/watch?v=b-b1cKBsG1I`

[2]CK1: `http://www.youtube.com/watch?v=wkTi8PfKFhY`, CK2: `http://www.youtube.com/watch?v=E5sMRIY_D5o`

In CK, a chord triggers an immediate response from the agent system through a reset of the pulse generators of each agent. This ensures that the agents produce notes in sync with the human playing. With legato playing, pulse generators are not reset – triggering only happens in transition from no key to one or more keys pressed.

For the performer, a significant difference between CK and CS is that in CK, I play *with* the system, not only *on* it. Every key makes a sound, and affects the system parameters. This makes me listen in a different way than when performing with human performers, or normal solo piano improvisation. I listen for things that work with what I actually play, structurally and harmonically, at the same time as I, with every note I play, affect the behavior of the system. This is truly balancing on a tight-rope. Still, it does not feel like I am playing with another musician. The experience it more like performing within a dynamic system that flexes, bends and changes as I play. The effect of any change is immediately felt, and the interaction is very evident, yet intricate and obscure in its inner mechanics. Focus is on flow and presence more than on cognitive analysis. As a performer, I feel that I have an effect, but I do not have time to think about how it works. The music emerges from these interactions in a very unpredictable yet organic way. Other performers may experience this differently. The system has so far primarily been used by me.

An interesting observation came from listeners accustomed to my acoustic piano improvisations – they say the CS system sounds like me playing. Every single design choice in a generative system has an aesthetic impact, even minute tweaks might affect musical results noticeably, sometimes radically. It is an interactive process of realizing a preliminary version of the idea, listen to it, apply tweaks, while possibly embracing unexpected results, promoting them to become integral parts of the concept of the work. In this process, aesthetic preferences of the designer are crucial. So, it may not be surprising that the system sounds like me. Maybe more surprising is that this kind of playing can be codified and controlled in such a compact way.

3.1 Systemic improvisation

The way this system is intimately coupled to the playing of the performer makes it feel like I am performing with an autonomous, very responsive co-improviser with very big ears. The system reacts to nuances in my playing, often in unpredictable but interesting ways, which gives material for further improvisation. This in turn triggers new responses, etc., in an iterative exploratory process. From this interplay between human performer and the agent system emerges very characteristic and rich music. The output cannot sound in any arbitrary way, because the system has potential to produce a certain kind of textures with regards to rhythm and pitch patterns. Still, this material is quite varied, and there is plenty of room for exploration. Even after a number of performances and recordings with the system, I do not feel it is near to be exhausted. However, the very characteristic style of the output makes me regard it as an artistic work, and not as an generic instrument or tool.

Still, there is no predetermined form or time-line. The form follows from the interactions between player and the system, and the "style" of the result is recognizable and characteristic. I call this kind of music, emerging in a characteristic way from such interactions a *systemic improvisation*. The system provides a mode of interaction that results in specific musical patterns on many levels. It can be considered an artistic work, because it is a direct consequence of the design of the system, but it is also dependent on the performer who plays it. There are many parallels with the genre of *game-pieces*, as used by, e.g., Christian Wolf and John Zorn. In their works, improvisers play within a framework of predetermined interaction patterns, while there is still room for their individual style, forming an artwork that can be realized in many different ways.

4. CONCLUSIONS

The presented generative live algorithm is simple in structure, but complex in results, thanks to internal feedback. Such feedback systems are notoriously difficult to control, but since all parameters are played by ear, the performer can respond to its playing, change direction when needed, or enhance what is going on. Through these fast interactions between man and machine, characteristic music emerges. The project shows a new approach to the old co-improviser paradigm, by allowing simultaneous physical gesture and parameter control of a complex generative system – while interacting musically with it through the very same piano playing. Playing *on* and *with* the system at the same time.

5. ACKNOWLEDGMENTS

This research was carried out within the Creative Performance project, funded by the Swedish Research Council.

6. REFERENCES

[1] P. Dahlstedt. Creating and exploring huge parameter spaces: Interactive evolution as a tool for sound generationwan. In *ICMC*, pages 235–242, 2001.

[2] P. Dahlstedt. Dynamic mapping strategies for expressive synthesis performance and improvisation. In *LNCS 5493*, pages 227–242, 2009.

[3] P. Dahlstedt. Autonomous evolution of piano pieces and performances with ossia ii. In E. R. Miranda, editor, *A-Life for Music: Music and Computer Models of Living Systems*. A-R Editions, 2011.

[4] P. Dahlstedt. Pencil fields: An expressive low-tech performance interface for analog synthesis. In *Proceedings of NIME12, Ann Arbor, MI*, 2012.

[5] A. Eigenfeldt. Multi-agency and real-time composition: In equilibrio. In *Proceedings of the Toronto Electroacoustic Symposium*, 2009.

[6] C. Goudeseune. Interpolated mappings for musical instruments. *Organised Sound*, 7(2):85–96, 2002.

[7] P. Laine. Feedback in musical computer applications. In *Proceedings of the International Computer Music Conference (ICMC), Beijing, China*, 1999.

[8] P. Laine. Motor neuron based virtual drummer. In *Proceedings of the International Computer Music Conference (ICMC), Beijing, China*, 1999.

[9] D. J. Levitin, S. McAdams, and R. L. Adams. Control parameters for musical instruments: a foundation for new mappings of gesture to sound. *Organised Sound*, 7(2):171–189, 2002.

[10] J.-C. Risset. From piano to computer to piano. In *ICMC*, 1990.

[11] N. Sjösvärd. Dialog. Composition for Disklavier and live electronics, 2012.

[12] D. Van Nort and M. Wanderley. Control strategies for navigation of complex sonic spaces. In *NIME07, New York, NY, USA*, 2007.

[13] M. Wainwright. Skyland mountain blues. Composition for Disklavier, electr. and tape, 2011.

[14] D. Wessel and M. Wright. Problems and prospects for intimate musical control of computers. In *Workshop in New Interfaces for Musical Expression, Seattle*, 2001.