Algorithmic Interfaces for Collaborative Improvisation

Shelly Knotts

Durham University, Durham, UK
michelle.knotts@durham.ac.uk

Abstract. The paper explores three different systems designed by the author which take critical approaches to algorithmically mediating collaborative improvised performance taking place between (i) co-located performers interacting with a graphical interface, (ii) performers with diverse musical practices performing together telematically, and (iii) co-located live coding performers. The three performances have different approaches to algorithmic mediation. In (i) performers work directly with the same sound sources via a GUI which is mediating each performer's input individually through restricting interface resources, in (ii) 'musical consensus' is deduced through machine analysis of the inputs assuming that the 'musical average' is the point of consensus, and in (iii) algorithmic agents with pre-defined musical preferences direct the mediation. The paper describes the initial considerations, design and implementation, and evaluation of the three systems, as well as a discussion of the issues raised by the use of such systems.

Keywords: collaboration, improvisation, algorithmic, generative, machine listening.

Introduction

Network music systems have been concerned with social organisation of performances since the late 1970s when network music pioneers The League of Automatic Music Composers (later became The Hub) began experimenting with building their own network structures for passing musical information and directions (Bischoff et al 1978). The ability to exchange and repurpose multi-modal data between performers and open new communication channels has driven network music system designers to imagine new structures for the temporary community of the musical ensemble.

Cross (2003) points towards music making as a “safe” domain for testing cultural theory. We might consider then, that network music systems, which utilise technology integral to contemporary social interaction and community formation, may interrogate the nature of these interactions and structure the exchange of available communication channels – audio, visuals, text, data – in ways that reflect and critique the nature of online interaction.

With the development of social networking, interface design and algorithms have become integral to our social interplay with data collection, content personalisation, and behaviour prediction, mediating the potential interactions we have and e.g. character limits, 'like' buttons, and even Instagram filters informing the content of our interactions. Algorithmic mediation and content creation, and interface design in network music systems are central topics explored through the systems discussed in this paper.

The section which follows describes briefly the design of *Controller*, *Union* and *Flock* and the specific aesthetic and technical considerations which impacted the development of each system.

The Projects

Controller

Controller is a GUI based project which aimed to design an interface for group improvisation, where performers’ actions are mediated by a central control mechanism which modifies their ability to contribute to a group performance. To facilitate this interaction a simple interface was designed with which to control both a shared sound space and the social structures within the group, thereby requiring the players to act both as performers in modifying the sound, and as conductors in changing the contribution of other players to the modification of sound. This group action is subverted by the mediating control mechanism which modifies throughout the piece the performers' abilities to contribute to both the sound and social aspects of the piece.

Interface Design

Controller consists of a simple interface on each performer's laptop. The interface is used to send control values over the network to set parameter values of a shared sound space. The interfaces are linked over the network in such a way that changing the setting of an interface element on one laptop changes the setting accordingly on all other laptops.

The interface itself consists of three basic types of control: sliders which can be used to control elements of the shared sound space; buttons which switch on and off the sliders of other performers; and knobs which effect the number of control data messages per second sent from a player's slider to control the shared sound space.

Structure

The structure of the performance is shaped by algorithmically setting the visibility of interface elements for each of the performers through a combination of 'random' allocation and 'special events'. Any one performance always follows the same basic structure of beginning with allocating each player just a small number of sliders, to increasing the number of sliders and rate of change of the allocation of interface elements. Buttons and knobs allowing performers to interact with the visibility of other performers' controls and therefor the social structures of the piece appear in the second half of a performance.

Aesthetic Design Considerations

There were several aesthetic considerations in play in the design of the performance interface relating to how the player interacts with the performance and enforcing particular social structures, some of which are discussed below.

The choice of interface elements was designed to make it easier to interact with the sound space than to interact with the social space. Therefore small buttons and knobs were used to control the social space and larger sliders were used for controlling the sound space thus asserting that the primary role of the performer is that of interacting with the sound and perhaps reinforcing social changes as subversive behaviour.

The status of other performers (i.e., how many controls they have relevant to other players) and therefore the overall social hierarchy is not revealed to performers. Neither is any indication given as to whether changes to the availability of elements on the performers' interface are the result of computer or performer action. However an audience graphic has been developed which gives audience members an overview of all performers' interactions with their interfaces.

The level of difficulty of performing with the interface should vary over the course of the piece, in essence, providing another way of controlling the interaction of players with the interface.

Conclusions

Laptop ensemble composition provides many possibilities for facilitating democratic interactions among performers and many network music pieces deal with shared sound spaces or shared controllers in order to facilitate collective action. *Controller* comments on this democratic potential by providing an interface which varies during performance to change the level of control participants have. Further, the work brings to the foreground for performers and audience the complex underlying group dynamics of interaction, with an ongoing negotiation of 'who controls what'. The piece extends the notion of 'composing democratically' for laptop ensemble (Knotts 2015) into the arena of 'composing democracy', that is, creating a musical structure out of shifting group dynamics, with political action at the forefront of the compositional design.

Union

The development of *Union* was the result of working with two different female-only laptop ensembles in 2015. The author worked with FLO (Female Laptop Orchestra) from 2014 to October 2015, and later co-initiated OFFAL (Orchestra For Females And Laptops), as a specifically telematic ensemble. Knotts (2014) surveyed the general characteristics of laptop ensembles and found only one female-only group, whereas many groups were male-only. *Union*'s development and the later formation of OFFAL arose out of the desire to form a large ensemble of improvising laptop musicians who are women[1]. Strategically the most practical way to develop a large ensemble of female performers was to develop a system for telematic collaboration. This facilitated the inclusion of many more women without the need for large funding grants to rehearse and perform. Further considerations were that in order to be inclusive to as wide a range of participants as possible the system needed to use free and open-source software, allow performers to collaborate with minimal intervention to their normal performance setup, and should work on non-institutional and low-speed and -bandwidth internet connections. These technical considerations defined many of the starting points of the project. Additionally the author had experience of audio streaming setups through several performances as part of GIASO (Grande Internationale Audio Streaming Orchestra).

While co-located network music systems offer the opportunity to develop alternative ensemble structures through designing the structure of additional information exchange (musical data and textual), telematic systems can provide interesting structural challenges due to the dislocation of performers from the site of the performance and from each other. In addition, inherent network delay may create varying levels temporal asynchronicity – depending on the technology and quality of network used. These issues can lead to social and musical disconnect as temporal asynchronicity and lack of visual feedback means performers can less easily perceive the actions of the other performers in relation to their own actions. In telematic scenarios chat clients can provide a site of social cohesion, a means to describe the remote locations, performance site, and audience interaction, and to give textual feedback and discuss creative aims.

Basic Setup

The audio from all participating performers is sent using streaming software to an Icecast server where all eight streams are then accessed by the system laptop. SuperCollider's SCMIR library is used to analyse the spectral content of the streams and mix according to similarity. The mixed version of the audio streams is sent directly out to the loudspeakers in the concert space in 8 channels and back to Icecast as a stereo mix, where online audiences can listen to the performance using software such as VLC, and performers can also monitor the final mix. Some basic testing revealed that this setup results in ca. 5-20 seconds delay between the individual performers sending their sound to Icecast server and hearing the same sound back as part of the final mix.

Developing the Algorithm

When performing with GIASO, there is always a performer in the performance space who is responsible for the final mix of the incoming streams. This solves many issues of asynchronous performance as there is a final 'gatekeeper' who can curate which streams currently sound most coherent together. However this raises some interesting considerations as clearly this act of mixing mediates the musical flow of the performance by picking and choosing whose sound gets heard by the other performers. Many different potential versions of the same performance are never realised due to the mixing decisions of this key performer, as their curation impacts not only the current moment, but the future decision of performers in how to react to the current sound mix. Additionally, the human mixer is not free from bias, and memory limits mean the ability of the mixer to mix 'fairly', i.e., so that all performers have a relatively equal amount of participation in the performance, is limited. Therefore we can clearly see that a performance hierarchy exists whereby the mixing performer has a large amount of power to shape the structure and musical narrative of the performance. This analysis served as the basis for designing an algorithmic mixer, with 3 distinct tasks: (i) to ensure a sound musical

---

[1] Including any female-identifying, or gender non-binary persons.

structure at the macro-level of the performance; (ii) to ensure equality of participation among performers; (iii) to use musical 'consensus' as a basis for shifting the power balance of the mixing task back towards the performers.

The *Union* algorithm uses machine listening techniques to detect similarity between the streams and attempts to balance 'consensus' against 'fairness'. The basic algorithm mixes the stream with the most 'average' audio features the loudest, scaling the other stream's amplitudes between 1 (most average) and 0 (least average). A higher level algorithm tries to ensure musical flow by moderating the density of the performance – every section has *n* streams which will be audible and every stream after the first *n* most average streams has an amplitude of 0 for that section. Each section has one more or one less stream than the last, with a slight tendency towards building up the density. Section breaks are made when the system switches to giving the *least* average stream the loudest amplitude. A final step in the algorithm balances the above with trying to ensure that all players get approximately the same amount of time 'on air'.

In performances with the system, a projection is used to show the online chat interactions between performers, giving audience members an insight into the musical decisions of the players, and their reactions to the algorithmic intervention, revealing to audience members the humanising elements of the musical interaction. In addition a visual aspect was developed which shows a (highly processed) photo of the currently audible performers. This seems to be relatively successful in showing how the algorithm is functioning to audience members and linking sound to specific performers.

Flock

Flock explores flocking mechanisms in network structures as a means of managing collaboration in a live coding performance. Loosely modelling the behaviour of bureaucrats in their interactions with self-regulating political systems, the three performers engage in a live coding election battleground, hoping to win votes from an artificial population. The more votes a performer wins, the more prominent in the final mix that performer’s audio signal will be.

Flock is a system of voting agents who each have musical preferences. In a similar way to Union the audio of several performers is analysed using machine listening techniques and at regular intervals the agents vote for whichever stream has the closest audio feature values to their preference. The audio is mixed accordingly, i.e., if a performer gains 40% of the votes their audio level is set to 0.4. In an attempt to more closely model real world electioneering, the agents are connected via a bi-partite network and their preferences are influenced by the outcome of each round of voting and the preferences of other agents they are connected to. The performance uses live coding as an analogy for political rhetoric and policy writing with the updating of a running process to get closer to AI preferences as similar to the way that politicians rescript their policy proposals in response to opinion polling.

SuperCollider’s JITlib was chosen as the language of algorithmic politics as SuperCollider is a reasonably neutral language, which places relatively few stylistic limits on performers, allowing them to form their own musical manifestos. Performers will develop their framework code in rehearsals beforehand, allowing them to form individual musical election strategies, before making their policy proposals (in musical form) to the artificial population in performance.

The voting mechanism itself is based on Rosen’s work on flocking in bi-partite decentralized networks (Rosen 2010). Rosen proposes that the most efficient and successful means of managing large group collaborations is through decentralised self-regulating networks. He suggests that these networks can maintain their decentralised non-hierarchical organisations through flocking mechanisms.

In *Flock* the network is made up of two types of node: feature trackers (using the SCMIR library in SuperCollider); and AI agents (who have preferences and voting rights). The feature tracker nodes hold information relating to the current feature states of the input audio from the performers (one node per feature). The agent nodes each have a profile, which includes ideal state preferences for each feature which is being tracked, and values denoting how autonomous they are – i.e., how strongly they are influenced by neighbour nodes. These features combined define the position of the node in the network and how it moves within the flock.

Performer feedback on the system includes that the system exerts a pressure to perform all the time in order to win votes, and that musical choices are influenced by the voting of the agents, i.e., a good strategy to win votes is to play similar sounds to whichever performer is currently leading. However, the flocking of the agents means that it's not possible to predict exactly how agents will react or move within the network and preferences change radically over the course of the piece, so the performers must also react to changing tastes of the agent population.

Conclusion

The systems described explore possible synergies between the dynamics of improvisation in music ensembles which use network technology to exchange musical and social data, and the dynamics of online social networking which is fundamentally mediated by algorithms and interface design. The three pieces discussed each explore the dynamics of human interaction when data collection and algorithms are used to modify or moderate this interaction, and algorithms subvert or enable particular power-dynamics within the group.

Magnusson (2009) points to the political, ethical and aesthetic nature of technology and how technology design structures human action and interaction, particularly software technologies, which are capable of reflecting the thought structures of their designers, necessitating a reflective responsibility from the ethical programmer. This is of prime concern in designing socially-aware systems which mediate performer interaction, and requires critical reflection and active acknowledgement that a value system is percolating through to the performance system. The impact of the system on performer agency should always be at the forefront of the design choices.

Relating to deliberative democratic ideals which resist consensus and the desire to avoid traditional composer-performer hierarchies, an important goal in my research into interface design was to not impose strict musical choices on performers, and therefore to have somewhat musically neutral design choices, which focus on performer interaction rather than content and musical structure.

Consensus is also an interesting avenue of exploration in relation to improvised performances, particularly in telematic contexts. The temporal nature of improvisation and the imperfect communication channels of audio and visual mean that immediate action and reaction without conferral is the standard route available to the unfolding of the performance in free improvisation. Musical dialogue, harmonious or antagonistic, directly drives the collaboration. Network music systems with the ability to collect data and provide text based communication in-performance, as well as the possibly controversial ability to restrict access to technical resources, allows us to investigate consensus in the context of musical interaction. Obvious arising questions which are actively explored through the design process include, how can we determine musical consensus and is consensus always desirable. The approach taken in the systems above was to develop algorithmic methods of deducing a form of consensus, while acknowledging the system designers' non-neutrality and the impact they therefore have on the group dynamic.

References

Bailey, Derek. 1992. *Improvisation. Its Nature and Practice in Music*. New York: Da Capo Press.

Bischoff, John, Gold Rich, and Horton, Jim. 1978. “Music for an Interactive Network of Microcomputers.” *Computer Music Journal 2(*3): 24 – 29.

Brown, Chris. and Bischoff, John. 2005. "Computer Network Music Bands: A History of The League of Automatic Music Composers and The Hub" in A. Chandler and N. Neumark (Eds.), *At a Distance: Precursors to Art and Activism on the Internet*. Cambridge: MIT Press.

Cross, Ian. 2003. "Music as a biocultural phenomenon." A*nnals of the New York Academy of Sciences 999*(1): 106 – 111.

Knotts, Shelly. 2015. "Changing Music's Constitution: Network Music and Radical Democratization." *Leonardo Music Journal 25*: 47 – 52.

Knotts, Shelly, and Collins, Nick. 2014. "The Politics of Laptop Ensemble: A Survey of 160 Laptop Ensemble and their Organisational Structures.". Proceedings of the 14th International Conference on New Instruments for Musical Expression 14: 191 – 194 (July).

Magnusson, Thor. 2009. "The Phenomenology of Digital Musical Instruments".

Murnighan, J. Keith, and Donald, E. Conlon. 1991. "The dynamics of intense work groups: A study of British string quartets.". *Administrative Science Quarterly 36*(2): 165 – 186.

Rosen, Devan, Jang Hyun Kim, and Yoonjae Nam. 2010. "Birds of a feather protest together: theorizing self-organizing political protests with flock theory."Systemic practice and action research 23(5): 419 – 441.