Swarm Granulator

Tim Blackwell[1] and Michael Young[2]

1 Department of Computing, [2]Department of Music,
Goldsmiths College, University of London, New Cross, London SE14 6NW, U.K.
{tim.blackwell, michael.young}@gold.ac.uk

Abstract. This paper describes a Swarm Granulator, a new application of particle swarms to sound synthesis. Granulation, an established technique in sound synthesis, depends on many parameters which are non-intuitive and hard to control from a human perspective. It is proposed here that a particle swarm can organize these parameters and produce musically interesting and novel timbres. A crucial element of the system is the self-organization of grain parameters around attractors which themselves represent musical events and textures in an external environment. This means that Swarm Granulator is interactive, and not merely reactive.

1 Introduction

Swarms, flocks, herds and shoals are natural systems that are remarkable in many ways, not least for their properties of self-organization. Killer bees chase and swarm around an unfortunate 'target', and termites exploit stigmergy (response at a later time to local environmental modification) and build elaborate nests. Starlings, fruit bats and herring all congregate in large numbers, developing spatio-temporal organization over large distances and long time-scales. It has been realized that spontaneous organization can develop without central control, but from interactions and a degree of positive and negative feedback [7].

Freely improvised music differs from composed music, or even music that accepts improvisation within an agreed structure (exemplified by some genres of jazz), by the degree of uncertainty that surrounds a performance. The performers (and indeed the audience) may have little idea of how the music will proceed before it starts [2]. Making improvised music is a social experience; social processes take precedence over traditional Western concepts like form and structure [8]. Such music is created spontaneously through the process of "becoming situated" [1] in which performers assume and cast roles, recognize and pursue shared goals and explore forms of interaction. Structure emerges as a consequence of these behaviors, (that is, from the 'bottom up'); players contribute musical material and interact with one another, establish relationships that are, to use Berry's terminology, complementary, counteractive or co-functioning (that is, relationships which are also evident in composed music) [3]. It is frequently the case therefore that 'free' improvisation can evolve structures, at least at a local level, creating the illusion of certainty, as if there actually is a conductor or a script. It has been proposed that self-organization is one mechanism for furthering the generation of spontaneous musical structure [6]; the computer intervenes and contributes to these collaborative processes.

It has already been demonstrated that a virtual swarm of particles can develop musically interesting 'improvisations' [4]. In Swarm Music, the particles move in a physical space. Particle positions are mapped onto sound-event parameters (such as event duration, loudness and pitch) and the swarm-like shape generated by the particles corresponds to a melody and, importantly, to an expressive performance of this melody. The result sounds improvisational rather than compositional because the fluctuations in the swarm shape produce ever-changing melodic and rhythmic variations. In this view, the organization of notes in a melody can arise from the (self) organization of particles in sound-event space.

However, Swarm Music is also interactive in the sense that it can both respond to, and initiate, changes in the musical environment. (A purely reactive system would only respond.) Following the inspiration from nature, the interaction is implemented as a series of targets or attractors, which represent modifications to the local environment. These attractors are parameterizations of musical events produced by human performers and by other swarms [4]. As the swarms organize around these attractors, musical ideas are generated which influence the improvisations of the musicians, leading to further attractor placements. Musical structure is generated stigmergetically by modifications to the environment of sound-event space.

Swarm Music is a MIDI-based system. It only provides sound-events; it does not specify how the events should actually sound. MIDI, like Western notation, is limited to the confines of the pitch-rhythm "lattice" [13]. The "lattice" represents music's conceptual confinement to the traditional hierarchy of musical 'notes', phrases, fixed instrumentation and so on. Pitch and duration are emphasized at the expense of the many other characteristics of a sound event, particularly timbre and its morphology. Musicians are acutely aware of these characteristics; a considerable part of their training is after all spent at developing instrumental control, and much of the expressive quality of music arises from timbral manipulations. Unsurprisingly, it is a key feature of freely improvised music, where the focus is often on expressive gesture, texture and the exploration of timbre. These concerns are shared by composers working with electronics and computers; there is a widely accepted aesthetic which rejects the pitch-based "lattice" as the only basis for musical organization.

It is pertinent, then, to consider if swarming can be used to develop music which explores these characteristics, and if self-organization can be used to relate timbres, (and even gestures and textures) to the impetus offered by external sounds. Artificial instruments can be constructed using various synthesis techniques [9]. The most appealing technique from a swarm perspective is granular synthesis because this technique provides a direct metaphor. Sound grains are packets of sound of very short duration; the asynchronous superposition of many of these grains produces a rich mass or 'cloud' of sound, whose characteristics are determined by a wide range of parameters [10]. It is tempting therefore to map the grains to particles, and the cloud to a swarm. Evolution of the cloud will be influenced by the attractors which themselves are a parameterization of the external sound events to which the system has access.

There are many ways an external sound could be parameterized as attractors in the physical space of the swarm, and many ways that particle positions can be interpreted as grain parameters. The next section explains the issues involved, and an interpretive model of interaction is presented in section 3. Section 4, which is an overview of the system, explains how the analysis of sections 2 and 3 is integrated in the final design. A brief evaluation concludes the paper.

2 Organizational Levels

The structural parameters of music, and their complex interrelations, form the basis of whole areas of scholarly endeavor. In this paper we can only briefly outline a working method for the swarm granulator. Music, whether composed or improvised, can be conceived in terms of a hierarchy of parameters, or organizational levels in which some properties (such as pitch, duration and timbre) are considered the most fundamental, and many others, (such as melody, harmony, rhythm, instrumentation) are accepted as complex, conceptual and historically/culturally specific [3].

One useful approach to understanding organizational level is based on perceptual and relative time scales [10]. The timescale closest to our immediate experience of sound is the sound-event (ministructural) level, a timescale around 0.1 to 10 seconds.

The granular process works at the grain-event (microstructural) level. Grains are measured in milliseconds; heard individually they may appear as clicks (that is, on the verge of timbral perception) or as longer fragments of recognizable source sound. We may on occasion consciously perceive that a sound comprises discreet events, but nevertheless it is not easily possible to measure the granular properties of a given sound.

The third relevant timescale is the mesostructural level; the level at which sound-events are experienced in relation to one another, rather than individually, as in the musical phrase, a melody or rhythmic pattern. Mesostructure can be considered as divisions of a higher level, that of musical form (the macrostructural level), or as a product of the lower sound event level. The interface between the sound-event and mesostructure is extremely hard to determine, especially outside the context of a given musical style. In fact in both free improvisation and electoacoustic composition, the alleged separation between these levels is itself explored and questioned, necessitating a alternative vocabulary characterized by terms such as "sound mass" and "cloud" [10].

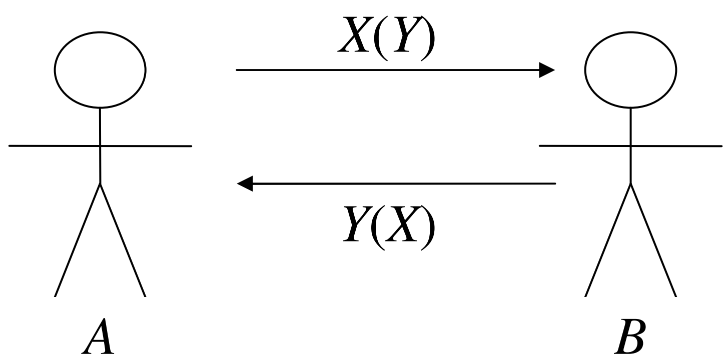
The essential difficulty faced in the design of Swarm Granulator is the decision about parameterization of the external sound, since this can occur at any level, and may even involve several levels. Most transparently, grain-event parameters could be extracted by wavelet analysis. The entire system would then be unified at this level, and it would be very interesting to explore self-organization upwards through the levels, from grains to notes, and from notes into form. (Swarm Music is a unified system, working entirely at the sound-event level.) It is not clear if wavelet analysis is a real-time technique; parameterization at the note-level is used in Swarm Granulator.

3 Interpretation

Interactive processes in musical improvisation are undoubtedly highly complex, functioning across different timescales/structural levels and subject to all manner of social and cultural influences, just as computer processes are subject to programming decisions. A simple model known as *interpretation* has been proposed in Swarm Music [4, 5]. Interpretation is a crucial function in Swarm Music and Swarm Granulator.

Figure 1 depicts interaction between two systems (human or silicon-based) *A* and *B*. System *A* (system *B*) is listening to an audio stream *Y* (stream *X*) emanating from *B* (system *A*). *A* (*B*) is also producing an audio stream *X* (*Y*). This picture, however, hides much. Human systems will be quite selective about which parts of the audio environment they will use to inform their own output, and this is desirable for silicon improvisers too. Interactivity merely implies that *A* is *influenced* by *B*, although *A*'s musical output will depend on many personal, hidden variables h_A which are unaffected by what he/she/it hears.

Fig. 1. A Simple Illustration of Interaction



The Swarm Music and Swarm Granulator systems both comprise attractors and swarms of virtual particles moving in a physical space. The attractors are parameterizations of the input stream, and the particle motions lead to parameterization of the external stream, whether at the note-level (Swarm Music) or at the grain-level (Swarm Granulator). The particles move autonomously, but they will respond to any attractors in their nearby environment, so that external events *influence* the swarm. Interpretation refers to the level-dependent rules for attractor placement, and on how the particle positions are used to modulate the output. In other words, interpretation is at the heart of system interaction.

A 'interprets' or attaches level-dependent meaning to the input *Y* in some way; this can be represented symbolically as $P: Y \rightarrow p$ where *p* represents some of the information which can be inferred from *Y*. If *A* decides to interact with *B*, then *A* must adjust her/his/its output in some way using this information, although this may not happen immediately, and the influence may be weak. In other words, A must memorize recent information p, and A can be quite selective about what elements of p to use. This process will be represented as $F(h, p)$: $x \rightarrow q$ where $F(h, p)$ is an internal process, dependent on hidden parameters h, that prepares output information q from internal states x. This formulation emphasizes that output is generated from internal processes which may depend only weakly on p, and can even continue in the absence of p. Finally, q modulates the output stream, which in Swarm Granulator is a stream of sound grains, using an algorithm Q, Q: $q \rightarrow X$.

4 System Overview

4.1 Swarming

Particle swarms ultimately derive from the virtual flocks of Reynolds' original animations [11], but the flapping animated 'boids' are replaced with structure-less point-particles in an N-dimensional 'physical' space. The particles change their positions by the application of simple forces or accelerations. In Reynolds original work, the accelerations are spring-like attractions towards the centre of mass of neighboring particles, a collision avoiding acceleration and a velocity matching term.

The particle swarm used in Swarm Granulator builds on the experience gained from Swarm Music. Particles are not stateless, but have a number of parameters which determine their interactions. A particle is specified by the set $\{\boldsymbol{x}, \boldsymbol{v}, \boldsymbol{p}\}$ where $\boldsymbol{x} = (x_1, \ldots x_N)$ and $\boldsymbol{v} = (v_1, \ldots v_N)$ are the particle position and velocity and $\boldsymbol{p}$ is the particle attractor (an N-dimensional vector). Five scalar parameters{c} determine the dynamics of each particle in the swarm; v_{clamp} is a clamping or limiting speed, q and m are for particle charge and mass, d_{core} is a small distance used to shape the inter-particle repulsion and d_{limit} is a perception limit. This perception limit is an extension of the perception limit for charged swarms, which was only relevant for the computation of inter-particle repulsion; a particle at $\boldsymbol{x}$ is only aware of other particles and attractors within a box $B_{limit}(\boldsymbol{x}) = [-d_{limit}, d_{limit}]^N$ centered on $\boldsymbol{x}$. These dynamical parameters $\{c\}$ determine particle motion, and hence ultimately sound output, independently of where (if anywhere) the attractors are, and correspond to the hidden parameters discussed in section 3. The swarm position $\boldsymbol{x}_{swarm}$ and swarm attractor $\boldsymbol{p}_{swarm}$ are defined to lie at the centroids of the particle positions and particle attractors respectively.

The particle dynamics are a set of update rules. These have been simplified from the rules used in Swarm Music and from other particle swarms. In particular, the spring constants determining the strengths of the attractions have been set at unity. This is because the parameter with the dominating effect on output is v_{clamp} [5].

The particle update rules for particle k, $k = 1, 2...M$ are given in Equations (1) – (5). Components of vectors along any direction are projected out by scalar products with the unit vectors $\boldsymbol{e}_i$, $i = 1 \ldots N$. Equation (1) calculates the accelerations which are

added to the velocity at iteration t-1 to form the updated velocity at iteration t, (2). The updated velocity is then added to the position (3). The linear spring-like attractions to swarm and attractor centers are preceded by a delta function, defined by equation (4), which ensures that these calculations are only applied if $\boldsymbol{x}_{swarm,\ t-1}$ and $\boldsymbol{p}_{swarm,\tau}$ are in the box $B_{limit}(\boldsymbol{x})$. The attractor is updated in real time τ and runs as a separate process to the particle update thread.
Equation (5) is a Coulombic repulsion between particles that are within the perception limit of each other, and is equal to a constant for separations less than the d_{core} and given by the inverse square law otherwise. Equation (5) sums up terms $\boldsymbol{a}_{k,l,\ t-1}$ which are the Coulomb repulsions between particle l and k, and equation (6) shows the calculation of a component of this term.
The Coulomb repulsion differs from the particle dynamics used in [5] because the spatial dimensions are decoupled. The update rules are merely N copies of a one dimensional dynamical system. Previously, the components were coupled through the Coulomb repulsion which was a function of the Euclidean distance $|\boldsymbol{r}|$ between particles. Dimensional coupling can still take place in Swarm Granulator, but it must be handled by the interpretative functions.

$$\boldsymbol{A}_{k,t} = m_k^{-1}\,[\delta(\boldsymbol{x}_{swarm,\ t-1}, \boldsymbol{x}_{k,\ t-1})(\boldsymbol{x}_{swarm,\ t-1} - \boldsymbol{x}_{k,\ t-1}) + \delta(\boldsymbol{p}_{swarm,\tau}, \boldsymbol{x}_{k,\ t-1})(\boldsymbol{p}_{swarm,\tau} - \boldsymbol{x}_{k,\ t-1}) + \boldsymbol{a}_{k,\ t-1}] \quad (1)$$

$$\boldsymbol{v}_{k,t} = \boldsymbol{v}_{k,\ t-1} + \boldsymbol{A}_{k,\ t} \quad (2)$$

$$\boldsymbol{x}_{k,\ t} = \boldsymbol{x}_{k,\ t-1} + \boldsymbol{v}_{k,\ t} \quad (3)$$

$$\delta(\boldsymbol{y}, \boldsymbol{x}) = \begin{cases} 1 \text{ if } \boldsymbol{y}_{k,\ t-1} \in B_{limit}(\boldsymbol{x}_k) \\ 0, \text{ otherwise} \end{cases} \quad (4)$$

$$\boldsymbol{a}_{k,\ t-1} = \sum_{l=1, l\neq k}^{M} \boldsymbol{a}_{k,l,t-1}\,\delta(\boldsymbol{x}_k, \boldsymbol{x}_l) \quad (5)$$

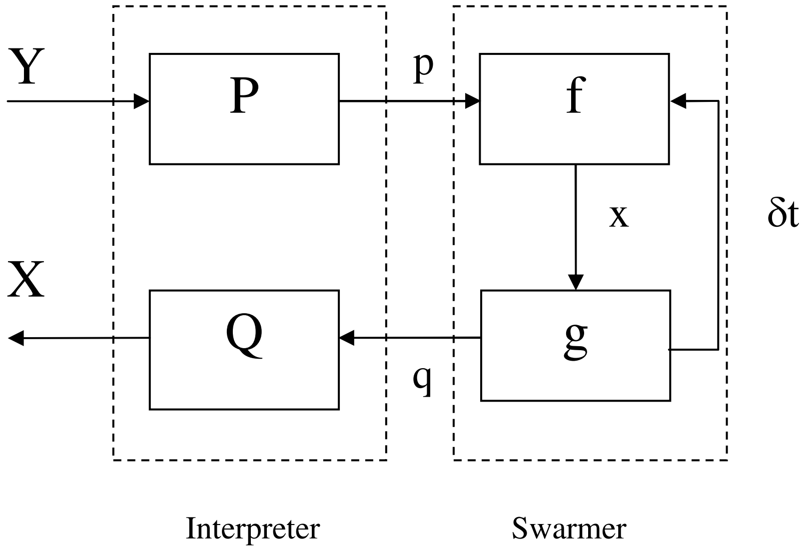
$$a_{k,l,t-1} = \begin{cases} \dfrac{Q_l Q_k}{|x_{core}|^2}\dfrac{x_{k,t-1} - x_{l,t-1}}{|x_{k,t-1} - x_{l,t-1}|}, & |x_k - x_l| < d_{core} \\ \dfrac{Q_l Q_k}{|x_{k,t-1} - x_{l,t-1}|^2}\dfrac{x_{k,t-1} - x_{l,t-1}}{|x_{k,t-1} - x_{l,t-1}|}, & \text{otherwise.} \end{cases} \quad (6)$$

4.2 Interpreter and Swarmer

This interpretive model forms the basis of Swarm Granulator which is comprised of two systems running on different computers, an interpreter and a swarmer. The interpreter is responsible for the listening and modulating functions P and Q, and the swarmer implements F. A diagrammatic overview is shown in Figure 2.

P parses the input stream into a series of *N*-dimensional parameters $\boldsymbol{p}_\tau$ (no level implied). The $\boldsymbol{p}_\tau$'s are sent to the swarmer where they are stored in a buffer. They are stored here for a holding time (which depends on the buffer size and the rate of flow of information into the buffer) after which they are placed in the physical space of the swarm as attractors. The buffer is a simple implementation of a memory and is important for stigmergetic interactions which are not instantaneous.

Fig. 2. Block view of Swarm Granulator



The swarmer splits *F* into two processes, *f* and *g*. *f* is a particle update function and *g* is a function which determines timing. Each particle *k* in the swarmer, $k = 1\ldots M$, is updated in turn in a continuous loop. Immediately after particle *k* has been updated, the swarmer pauses for a time interval δt_i. The index *i* is a counter which is incremented by one at each particle update. The particle update can be expressed formally as

$$\boldsymbol{x}_i \quad = \quad F(\boldsymbol{x}_{i-M}\ldots\boldsymbol{x}_{i-1},\{\boldsymbol{p}_\tau\},\{c\}) \qquad (7)$$

where *M* is the swarm size. Equation (7) is a formal statement of the rules (1) - (6) where $\boldsymbol{x}_i$ is the position of particle *k* at iteration *t*. Particle velocities do not appear in the swarming function *f* because they can always be constructed from the $\boldsymbol{x}_i$ using $\boldsymbol{v}_{k,t} = \boldsymbol{x}_{k,t} - \boldsymbol{x}_{k,t-1} = \boldsymbol{v}_i = \boldsymbol{x}_i - \boldsymbol{x}_{i-M}$. Equation (7) shows that the swarm can be replaced by a state machine whose current state $\boldsymbol{x}_i$ depends on its *M* previous states $\boldsymbol{x}_{i-M}\ldots\boldsymbol{x}_{i-1}$ and is influenced by the inputs $\{\boldsymbol{p}_i\}$ and parameters $\{c\}$.

A second function *g* extracts event information δt_i and $\delta t_{i,\ event}$ from $\boldsymbol{x}_i$ where δt_i is the time between successive events and $\delta t_{i,\ event}$ is the duration of this event. The output

from the swarmer is a stream of 'event' parameters (no level implied) $\boldsymbol{q}_0$, $\boldsymbol{q}_1$, $\boldsymbol{q}_2$… at real times T, T+δt_0, T+δt_1,… Two components of $\boldsymbol{q}_i$ contain timing information and the remaining N-2 components are real numbers.
The interpreter receives $\boldsymbol{q}_i$ and prepares output using Q. From the perspective of the interpreter, the swarmer merely transforms the input parameterization into modulating numbers $q = gf(p)$. The job of the interpreter is to listen at some structural level (P) and respond at the same, or different level (Q). The interpretation can be 'transparent' with $P = Q^{-1}$ (and this is the case in Swarm Music which operates at a single level) but this is not the only option. All that matters is that the interpretative functions P and Q are *transparent enough* for interacting humans to grasp and use during performance (a similar point applies for human-human interaction).

4.3 Granulator

The overall system has three modules; interpreter, swarmer and the granulator which is the actual sound engine. In granulation, or granular synthesis, grains are generated by multiplying an envelope (window) of given amplitude, duration and shape with a waveform. The simplest approach would be to employ a Gaussian envelope and sine tone waveform of a given frequency. Other envelope shapes are feasible, as are more complex waveforms (for example, derived from sampled audio, as in our case). Synthesis is achieved by iterating grains either synchronously or asynchronously. The result is a stream of sound with potentially very diverse timbral characteristics. Many grain-event level parameters affect these perceptual features; the diverse approaches to this technique are explored in detail by Roads [10].
The granulator is implemented using Max/MSP, with objects from a 'granular toolkit' [20]. In our current implantation, the interpretative function P operates at the sound-event level and Q operates at the grain-event level, and a transparent mapping is made from extracted parameters to grains. Specifically, P extracts four sound-event parameters: pitch, amplitude, duration and duration between successive sound-events. Q uses q to determine grain-event pitch, amplitude, duration, time between successive grains, and grain attack and decay time. The swarmer, therefore, operates in $N = 6$ dimensional physical space; attractor components in the dimensions representing grain attack and decay times are fixed. The grains are shaped with a Hanning window and the point of entry in the buffer of continuously sampled audio is set by the operator.
Three simultaneous grain streams are used. This means that there are three swarmers, each with different parameter settings $\{c\}$ (which can be altered by the operator) sending parameter streams to three granulators. The granulators and interpreters are implemented on a 500 MHz Apple G4 and the swarmers, written in Java, run on a 1.7GHz pc. The two machines communicate using our own implementation of the Open Sound Control protocol for Ethernet communication [21].

6 Evaluation and Future Developments

Swarm Granulator, which is still essentially a prototype of a more complete system, was nevertheless ‘tested’ at two recent events. These two performances of Swarm Granulator have both involved interactions between the granulator and live musicians. The first event was at the Modular 2003 meeting, London College of Music and Media, Sept 11 2003, with the classical singer Robin Higgins. The second event was part of Big Blip, the Brighton (U.K.) Arts-Science festival, Oct 11th 2003, with L.E.G., an improvising ensemble comprising Mette Bille (voice), Panos Ghikas (violin) and Johannes von Weiszacker (cello). Some excerpts from the second performance will be made available on a website [17].
Each concert comprised a single spontaneously improvised performance, undertaken with minimal preparation (a technical sound check). The performances lasted a little over ten minutes. Ultimately the performances can only be evaluated subjectively, that is, according to personal aesthetic criteria, but they demonstrated that the Swarm Granulator is sustainable under real-life performance conditions. Two reactions follow:

It was a really successful performance. Although I work in algorithmic composition it is rare indeed that I experience a system with genuine musicality, so it’s always exiting when that happens. Andrew Gartland-Jones, Big Blip organizer.

The effect on the listener was one of fascination; surprisingly musical. Howard Moscovitz, performer at Modular 2003 [15].

We can also gain some insight into the experience of the performers through discussion.

-Did your contribution feel valued...did the machine support your contribution?
PG: *My contribution did feel valued. I felt the resulting sound was very interesting because the 'machine' complimented my 'real' sound.*
-To what extent did you feel 'directed' by the swam machine?
PG: *I felt there was a sense of ‘direction’ as strong as the one that can be felt during ‘human-only' improvisation.*
-To what extent did you feel you controlled/influenced the machine?
PG: *The extent of influence I felt was varied but mostly strong.*
- Did the machine give you ideas?
PG: *Yes, mainly structurally.*
- Was this a new musical experience?*...in what way*?
PG: *I am not sure if it was a new musical experience but it was the most impressive musical interaction I have experienced with a 'machine'*!

Also these comments from Mette Bille:
“*The free improvisation with the ensemble and* [Swarm Granulator] *worked fine, it was great that you could record some of the playing and then send it back as an extra*

element to play with or against. I didn't find any problems with the program as an extra element as it was easy to hear what was happening. So generally it was just a question of tuning in [to] *each other as musicians…I thought* [it] *worked well*."

These comments do not in themselves verify the system but they do motivate further work with Swarm Granulator. Future developments may concern parameter extraction and mapping. Our current approach to parameter mapping is essentially one-to-one and at two levels. Although it is arguable that literal parameter mirroring is neither necessary nor desirable, it is desirable to integrate further characteristics of the sound event level (e.g. timbre) and some elements of the meso-level (such as changes in dynamics, rhythmic or pitch-based patterns). It should be borne in mind that parameter mapping is a creative process in any context, and the exploration of parameter relationships in improvised performance constitutes an intrinsic element of the live, creative and quasi-social process. In other words, all we need to do is to be consistent, since the musicians will establish mappings intuitively or aurally by experiment.

References

1. Bastien, D. and Hostager T. Cooperation as Communicative Accomplishment: A Symbolic Interaction Analysis of an Improvised Jazz Concert. Communication Studies 43 (1992)
2. Bailey D. Improvisation: its Nature and Practice in Music. The British Library Sound Archive. London (1992)
3. Berry, W. Structural Functions in Music. New Jersey: Prentice-Hall (1976)
4. Blackwell T.M. Swarm Music. MSc dissertation, University College London. 2001.
5. Blackwell T.M. Musical Interaction with a Multi-Swarm. AISB 2003.
6. Blackwell T.M. Swarms and Self-Organized Music. Newsletter of the Society for Artificial Intelligence and Simulation of Behavior. (2003)
7. Bonabeau E., Dorigo M. and Theraulaz T. From Natural to Artificial Swarm Intelligence. Oxford University Press, New York (1999).
8. Lewis G. Interacting with Latter-Day Musical Automata. Contemporary Music Review 18 (1999) 1
9. Miranda E. Computer Sound Synthesis. Academic?? (2003??)
10. Roads C. Microsound. MIT press, Cambridge (2002)
11. Reynolds C. Flocks, Herds, and Schools: A Distributed Behavioral Model, in Computer Graphics. SIGGRAPH '87 Conference Proceedings. 21(4) (1987) 25-34.
12. Rovan, J. Wanderley, M. Dubnov, S. and Depalle P. "Instrumental gestural mapping strategies as expressivity determinants in computer music performance." Proceedings of the Kansei - The Technology of Emotion Workshop. Genova, Italy, 1997.
13. Wishart, T. Audible Design. Orpheus the Pantomine Ltd (1994)
14. http://www.crca.ucsd.edu/~msp
15. http://www.electro-music.com/article.php?t=473
16. http://www.nathanwolek.com
17. http://www.timblackwell.com