NN Music: Improvising with a 'Living' Computer

Michael Young

Music Department, Goldsmiths, University of London
New Cross, London, SE14 6NW. UK.

m.young@gold.ac.uk

Abstract. A live algorithm describes an ideal autonomous performance system able to engage in performance with abilities analogous, if not identical, to a human musician. This paper proposes five attributes of a live algorithm: adaptability, empowerment, intimacy, opacity and unimagined music. These attributes are explored in NN Music, a performer-machine system for Max/MSP that fosters listening and learning. Live improvisation is encoded statistically to train a feed-forward neural network, mapped to stochastic processes for musical output. Through adaptation, mappings are learnt and covertly assigned, to be revisited by both player and machine as a performance develops.

Keywords: Live algorithms, improvisation, performance systems, artificial neural networks.

1. Introduction

Advances in our understanding of machine intelligence, in areas such as music informatics, evolutionary computation and self-organising maps, open new avenues for creative performance systems in music. Such systems might collaborate in many musical contexts, not merely follow pre-programmed scores or depend on direct stimuli, but engage with performers at a commensurate level. This is the vision of the UK Live Algorithms for Music network, founded in 2004 by the author and Tim Blackwell[1]. A live algorithm is the function of an ideal autonomous system able to engage in performance with abilities analogous, if not identical, to a human musician [1]. Such an approach differs radically from the established paradigms of 'live electronics'; the computer-as-instrument (a tool that relies on human agency), or the computer-as-proxy, (a substitute for the 'composer' that implements pre-designed functions laid out in a musical score or rule set). A true live algorithm would offer a high degree of autonomy: capacities to invent, provoke and respond.

Live algorithms are most germane when there are opportunities for such behaviours, i.e. when there is creative, group interaction. In this scenario there is no 'top-

---

1 A collaboration between the Departments of Music and Computing at Goldsmiths, with the support of the Engineering and Physical Sciences Research Council.

down' control, no hierarchical human-to-human management analogous to user-to-computer control. In truly 'free' improvised music, structure and character – in so far as they are evident – are emergent properties, products of heterarchical group interaction, and not the product of pre-defined rules. Musical languages are formulated pragmatically, self-referentially and on-the-fly. Free improvised music offers a model for aspiring live algorithms and a challenging context in which one could be tested to the full.

Section 2 below discusses potential attributes of live algorithms that are far-reaching and intended only as a framework for discussion and future investigation. There are five attributes; adaptability, empowerment, intimacy, opacity, and (summatively) the unimagined, as proposed by the author [2], [3]. These concepts are explored practically in the performance system NN Music, which has been developed in a number of compositional guises, with instrument-, timbre- and concept-specific titles: *piano_prosthesis, cello_prosthesis,* and *au(or)a*. There is a technical explanation of this system in section 3.

2. Attributes of 'Living' Computer Music

Adaptability. This is the ability to acclimatise to a shared environment, demonstrable in changes of behaviour. A musical environment capable of change – and therefore to demand adaptation – is unlikely to be pre-determined by fixed rules, stylistic assumptions or other formal constraints. Adaptation is not necessarily conscious or intentional, even though performers may wish to communicate with a machine. Lewis's term emotional transduction, defined as a "bi-directional transfer of intentionality through sound" [4] establishes by implication that adaptations should occur in and of the medium itself, not via controllers, irrelevant gestural information or control data. However, Lewis's assertion that a performer's original intention – "emotional and mental" – can be preserved and then co-represented in the machine's response is open to question.

It can be argued that performers adapt to their shared sonic environment, not personally to one another. Stigmergy avoids the problem of personal intention and emphasises adaptation; this provides a model that is potentially valid for both human-human performance and human-machine collaboration. Stigmergy is the process in which an insect population self-organises through the adaptation of individuals to their environment, and initially described termites interacting with their environment. Individuals do not commune directly, even though the resultant phenomena – nests – can be extraordinarily complex and seemingly designed. Computer simulations of this have been and other self-organising behaviours are well established and they are evidenced in the application of evolutionary computing to music [5]. Stigmergy as a model avoids the problem of intentionality and machine cognition; consequently, it avoids the potential pitfalls of anthropomorphism. It proposes a flexible, dynamic and adaptable system capable of novel problem-solving: An effective model of human creativity and more specifically, of improvised computer music [6].

Musical performance, whether between humans or machines, might be regarded as a complex and dynamic self-organising system if individuals commune with the shared audio environment rather than directly with one another. This assumption ignores visual or other physical cueing, and emphasises listening.

Musical collaboration in a human/social context involves a continuous process of adaptation based on mutual listening. Goals are identified by group members – who actively assume and cast roles – in order to adapt to the changing audio environment. Goals might be attributable to 'supra-personal' social facts (such as norms of acceptable behaviour, actions consistent with expectations or requirements). Even so, an entirely new, shared history evolves as the cooperative experience develops. Players become aware of the appropriateness of their response to others' contributions and appraise their own ability to initiate behaviour from others. Such processes have been observed in jazz groups [7]. Arguably, all such social behaviours occur by proxy in the shared environment: they are adaptive and essentially indirect. Whether or not it might ever be possible for computers to have an intentional response is an open question. However, optimisation methods, available in evolutionary computation and machine learning, might present the affect of intention. If interaction occurs only between performer and the environment, and between machine and its environment, the products of adaptation might be regarded as equivalent, and equally significant. It is then, arguably, inconsequential whether the interactions depend on machine algorithms or human cognition.

Empowerment. This entails control over decisions that impact upon future experience. Decisions have a context: the properties and consequences of options, the strategies that might inform choices and the criteria for their evaluation. In creative practice, such as improvised performance, decisions do not have easily definable strategies or evaluative criteria, but a framework must be at least implicit. Established AI systems that are effective in adapting to environments and delivering pre-defined outcomes are not necessarily useful. For example, BDI (belief, desire, intention) systems implement a rule-base, respond only to knowable environmental measures and have clear pre-determined aims, even though they actively respond to input and output while running [8]. Such behaviours are potentially antithetical to the exigencies of creative performance.

Algorithms do not cognate – so they cannot make creative decisions – but they can produce non-arbitrary changes in state. Such changes can be instigated by non-linear dynamical systems; cellular automata, particle swarms, genetic algorithms and neural networks. The potential self-organising properties of these algorithms offer potential for novel problem-solving, invention and surprise. They do not necessarily achieve intended goals, but might find new ones. In musical performance, a non-arbitrary change of state is manifest as a 'decision' when it modifies the audio environment, even if this is the product of an adaptation. Consequently, the change is 'empowered' to demand a response from both human and machine participants alike; it has the affect of intention.

There is mapping problem. What structural and temporal features of music should be determined by a change of state manifest as 'a decision'? It is easy to find examples of generative music where state changes are applied to the very surface of music. For instance in evolutionary computer music, genotype (genetic code) and phenotype (characteristics) have been mapped schematically and literally. Such approaches are, in effect, simple sonifications of a data space exploration, which may be of potential use for scientific enquiry [9] but have limited interest as a means of creative production.

Creative decisions might reference the structural properties, and implicit methodologies of music-making, and offer new possibilities at these levels. Computer 'decision-making' cannot define context, but could engage with it. A common problem is time; algorithms function independently of time, so for music, a real-time clock must be imposed as a function of data sonification. It is unavoidable that contexts such as this – however fundamental and transparent – are established by the designer, in order to provide relations for creative behaviour. Eco's term, the *"field of relations"*, emphasises the finite nature of an open work's discontinuities and its field of possibilities [10]. These relations provide a framework for decisions. So, even though a single point of view is absent and there is some devolution of creative responsibility, this does not entail an *"amorphous invitation to indiscriminate participation"*. Neither, by extrapolation, does the absence of a point of view (algorithms do not cognate) necessitate a wholesale and literal transfer of state changes to the surface of the music, or to the framework and context for creative acts. Relations are underpinned by the capabilities of the machine system, the technical approaches and aesthetic attitudes of the designer and live player. It is through an interplay of all these relations that empowerment might be perceived.

Intimacy. This is experienced – or apparent – if there is a binding understanding shared by performers through informed listening and observation. This is a social process, but can be experienced in and through sound itself. A machine emulation of closeness and intimacy should attend to sonic experience, both in nuance and wider characteristics.

Technological devices that produce control data from a user's actions can only be receptive, not intimate. In music technology, the discourse around intimacy is really about responsiveness, i.e. emulation of a performer's physical interaction with his/her instrument [11]. A truly intimate relationship – as occurs between musicians – is learned, rather than provided, and is an experiential phenomenon within the sound environment. (At least during a performance, before or after is another matter). It is, though, genuinely interactive.

Intimacy suggests the psychological process of *"optimal flow"*; a goal-orientated, mental state that explores the limits of experience and expectation, obtaining pleasure in meeting challenges with appropriate skills [12]. It has been conjectured that the effectiveness of group collaborations can be evaluated with this measure [13]. A machine's contribution cannot be evaluated, of course, but a human performer, in his/her musical experience and interaction with the shared sonic environment, might infer that flow is occurring for all participants. This is particularly relevant when, for example by using neural networks, a machine can evidence prior learning and experience.

Opacity. This is a prerequisite for this flow, an avoidance of the naïve processes of cause and effect (and their frequent boredoms for players and audiences alike). Interactivity is a well-discussed term in computer music but its currency has become a little devalued. It is often equated with a one-directional transfer of information from user to machine; reaction, not interaction. A lack of opacity and uncertainty distances the performer from the machine. The relationship is then that of a familiar 'subject-to-object', which by implication denies the possibility of intimacy: *"…interactivity has gradually become a metonym for information retrieval rather than dialogue, posing the danger of . . . reifying the encounter with technology"* [4]. George Lewis offers Voyager's capacity for *"variation and difference"* as an alternative that avoids transparent and consistent input-output mapping, but still provides against the appearance of randomness.

A truly interactive system ought to offer an ambiguous and shifting balance between the reactive and proactive, and across the threshold of the apparently chaotic and the readily comprehensible.

Unimagined. The result of these attributes might be a 'living' computer music, an unimagined music, its unresolved and unknown characteristics offering a genuine reason for machine-human collaboration. If computers might extend, not parody, human creative behaviour, machine music should not emulate established styles or practices, or be measured according to any associated, alleged aesthetic. In living computer music the contributions of all performers involved – human and machine –have equal significance, but may not necessarily be equivalent. Such music cannot be imagined or reproduced.

Unimagined music, free from pre-defined rules or overt control, moves *"… toward a permanent discovery – comparable to a 'permanent revolution'."* [14]. Boulez refers to compositional method, the exigencies of musical form given a *"fluidity of vocabulary"*, and the consequent need to de-linearise temporal structure. However, a 'living' computer music might be even more apposite, permanently exploring all elements of its emergent language, and in real-time, not just in concept.

Freedoms, whether open to the player or to computer (e.g. by stochastic methods) might be better described as 'informalities'. Unimagined music is a technological "*musique informelle",* emergent and idiosyncratic; its coherence neither derived nor dictated. It *"discards all forms which are external or abstract or which confront it in an inflexible way, free of anything irreducibly alien to itself or superimposed on it"* [15]. Adorno's term is not synonymous with the informal and intuitive; it does not deny the potential for objective and measurable structural complexity. This approach is arguably apposite to free improvisation and the claims of its practitioners. In 'living' computer music, unpredicted acts of a performer, and implicit (i.e. virtual) acts of the machine should exemplify this objective complexity, but not through the simple sonfication of rules or sheer randomness. There should be a critical engagement between intended behaviours, an appraisal of potential behaviours and response to actual sonic realisations and their unfolding history. Ideally, there should be an integration of subject (performer) and organism (the musical system regarded as a whole).

2 NN Music: A Performance System

These five properties are addressed in the NN Music (Neural Network Music) performance system, which brings together a solo player with a computer to mutually interact by proxy in the sonic environment. Implementation is in Max/MSP, including the neural net external object op.fann.mlp by Olivier Pasquet[2]. NN Music has been deployed with a number of instrumental combinations under the titles *au(or)a*, *piano_prosthesis* and *cello_prosthesis*, the titles indicating a particular musical/compositional ethos. *Au(or)a* is intended for any solo instrument with Disklavier piano; the system produces MIDI data intended to emulate a virtual pianist (with the veracity of real-life instrumental sound offered by the Disklavier). The two *_prosthesis* works begin a projected series of pieces that bring together a specific instrument with a related (and transformed) library of samples and real-time manipulations. There is therefore an added dimension of digital synthesis. The common creative concerns in these works are **distance** (conceptual and physical distances, human-machine aesthetic differences) and **embodiment** (to explore the notion of 'prosthesis' an alternative to the user-interface paradigm for HCI [16]).

2.1 PQ*f*

NN Music system is best described according to the modular PQ*f* architecture proposed for improvising performance systems [6]. This simple modular structure offers an empirical algorithmic model of the process of listening and comprehension (P), rendering of performance (Q) and creative thinking (f) experienced by human performers. It offers a direct analogy for machine musicians, although there is no attempt at symbolic representation. For a machine, P is an audio analysis function, Q is a synthesis function and f(h) is some form of hidden, organising algorithm. P and Q interpret, and interface with, the sonic environment, and also communicate with the hidden algorithm: P obtains an analysis parameter set $\{p_0, p_1, \ldots p_n\}$ from an audio stream X, relaying these to the algorithm. Q generates a synthesis set $\{q_0, q_1, \ldots q_n\}$ from the output of the algorithm, which creates a new audio stream Y. This supposes that changes in state evidenced in f(h) are scheduled in real-time, subject to the contingencies of the inputs and outputs from and to the environment. X and Y may be considered either as distinct audio streams that contribute to the total sonic environment, or as two temporally spaced points on one continuous stream, depending on the musical application and hidden functions.

$$X \xrightarrow{P(fX)} \{p_0, p_1, \ldots p_n\} \rightarrow \boldsymbol{f(\mathbf{h})} \xrightarrow{Q(fY)} \{q_0, q_1, \ldots q_n\} \rightarrow Y\,. \quad \textbf{(1)}$$

There is a vast array techniques deployed for all three modules in this schema. Creative computing applications have explored many approaches to 'hidden' genera-

---

[2] Available at www.maxobjects.com

tive algorithms $f(h)$; often, non-linear dynamical processes that display complexity, capacity for self-organisation, and/or patterns perceived to have some artistic value. Cellular automata, genetic algorithms, particle swarm and flocking algorithms and self-organising maps might be classed as such. Although differing widely in approach, these sub-symbolic techniques can be distinguished from those that have recourse to a pre-defined, 'expert' rule-base. The latter approach constitutes a direct mapping from P to Q, i.e. an interface between performer and machine via a set of contingent rules in which all circumstances may be predicted in advance. Such approaches cannot be classed as 'live algorithms'.

Real-time Neural Networks. The feed-forward neural network allows unsupervised sub-symbolic learning and classification; the potential for self-organising, learning behaviours effective as an $f(h)$ patterning algorithm. The multilayer perceptron neural network is trained using a back-propagation error algorithm that minimises the error between required and actual outputs by gradient descent, given a set of pre-defined input and output conditions. As noted by Toivianen, this type of network benefits from a capacity for generalisation and tolerance to apparently unpredictable or contradictory data; consequently it is well suited to classifying analysis of improvised music, the input state comprising attributes of the improvisation [17].

In NN Music, audio analysis P and synthesis Q are mapped via a feed-forward neural network in real-time. The network adapts to attributes of the performance, and outputs synthesis parameters accordingly. This is an original application for feed-forward networks in that the training phase occurs during the musical performance, not prior to it. Training repeatedly recurs during performance, depending on the variance in musical behaviour measured from the human-produced sound. It compares the current behaviour to the history of classified behaviours in that performance, rather than simply to a set of previously defined classifications. This enables the system to adapt to the contingencies of a specific performance, and also, through the writing/reading of network weight files, to previous performances. There is capacity for immediate short-term learning and long-term memory, represented in the simple form of neural network mapping.

The knowledge of the trained network is opaque, embedded in its re-formulation of internal weights, and can only be ascertained through experimental enquiry. This affords opportunity for creative investigation, but only in the intimate moment of performance itself.

2.2 Analysis and Learning: P → f.

There are two analysis functions, P_{pitch} and P_{audio}. The first focuses on pitch; the implied harmonic characteristics of the improvisation (rather than just step-by-step note progression) and executes a harmonic function to extend the characteristics logically, providing a related, wider pitch resource. The second analysis, P_{audio}, is independent of this, and measures characteristics of the performance based on various audio descriptors to be used as inputs to the neural network.

Pitch analysis and generation. Figure 1 shows the pitch analysis function, P_{pitch}. Audio to pitch conversion produces a stream of data, accurate to the nearest quarter-tone, which is filtered by an *attentiveness* function; the probability that a pitch will be allowed to update the dynamic set S_{chord}, a list of most recently admitted pitches $\{x_0, x_1, \ldots x_n\}$. In current versions, $n = 6$. The filter is deployed dynamically, mapped from the mean onset density detected over an adjustable time Δt, so relative inactivity on the performer's part fosters more attentive machine listening. When the primary set S_{chord} is updated with a new pitch, a generative function, f_{gen}, recalculates ten other hexachords by cross-multiplying each pitch within the primary set. The resultant chords are identical, other than in their transposition, and each member of $S_{chord\text{-}set}$ contains at least one of the pitches from the original hexachord S_{chord}.

$$f_{gen}: S_{chord} \rightarrow S_{chord\_set} \qquad (2)$$
$$\{x_0, x_1, \ldots x_n\} \text{ X } \{x_0, x_1, \ldots x_n\}.$$

This method emulates the post-serial technique of chord multiplication, devised by Boulez (as, for example, identified in the 'L'artisanat Furieux' cycle of *Le Marteau sans Maitre* [18]). The difference in this instance is that this function continuously updates $S_{chord\text{-}set}$ in real time as new pitches are admitted. $S_{chord\text{-}set}$ is a dynamic pitch corpus, deployed as a resource for the synthesis function Q (explained below). The system adapts to the pitch content of the player (who may to decide to opt for a particular musical approach – e.g. freely atonal, modal etc.), providing a cohesive harmonic framework that is neutral and apposite to "non-idiomatic" free improvisation, as advocated by Bailey [19]. It also creates opacity, due both to the detailed statistical filtering of note admission and the complexity of f_{gen} itself, offering a challenging but comprehensible environment to which the performer in turn may adapt.

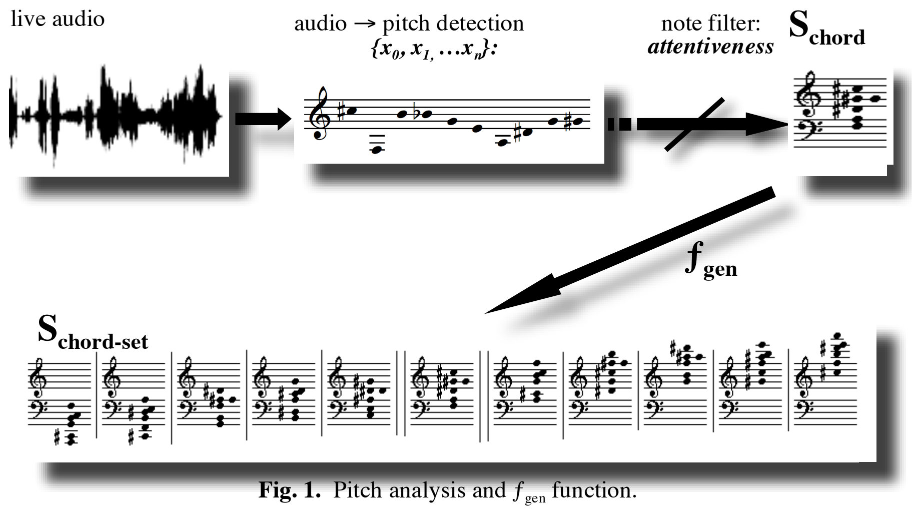

Fig. 1. Pitch analysis and f_{gen} function.

Audio analysis and training. The second analysis function, shown in figure 2 below, applies audio descriptors to the live performance (loudness, brightness, duration between events, sustained-ness, frequency etc.) with an analysis window of 50ms. It creates a dynamic performance state, S_{audio}, which is a statistical representation of the performance behaviour, measured over time, Δt, comprising the normalised mean ($\bar{x}$) and normalised standard deviation (σ) of all the descriptors, where 5s < Δt > 30s. The state S_{audio} might indicate a musical behaviour as follows:

- very loud dynamic: $\bar{x} \approx 1.$, $\sigma \approx 0.$
- intermittent bursts of rhythmic activity: $\bar{x} \approx 0.5$, $\sigma \approx 1.$
- low pitch: $\bar{x} \approx 0.$, $\sigma \approx 0.$

Rather than observing a simple stream of events, the analysis attempts to represent a musical behaviour in such broad terms: this is relevant to the exigencies of freely improvised music, although the analysis is, in itself, only indicative. It is adaptable however, as the individual descriptors in themselves are of less significance than the composite representation offered by S_{audio}.

The purpose of network A is to classify novel performance behaviours, as represented by S_{audio}, in order to acquire a library of learned states for future reference $\{S_0, S_1, \ldots S_n\}$. This learning is applied – while the improvisation continues and the network runs – to assess incoming states in comparison to those already known: the aim being to identify musical behaviours that are well defined and contrasting, so the network can respond effectively to a broad range of subsequent musical activity. To achieve this, the dynamic state S_{audio} is considered for retraining only if it satisfies the fitness function f_{fit}, a measure of the similarity of the current S_{audio} to all those previously learned. The function, found through experimentation, is represented as coefficient a, the sum of the mean and standard deviation of the absolute difference between the new state under consideration and a previously admitted state. This produces a list of values, $\{a_0, a_1, \ldots a_n\}$, where n is the number of already admitted states. If any value of a is greater than a predetermined threshold z, the new state is allowed to update the network, which is retrained on the fly; otherwise it is discarded.

$$f_{fit} : S_{audio} \rightarrow \{a_1, a_2, \ldots a_n\} > z \, . \tag{3}$$

In the current implementation, the threshold is set by the user; to be effective it must adjust to characteristic behaviours of both instrument and performer. The number of output nodes increases every time a new state is classified, $\{O_0, O_1, \ldots O_n\}$ representing an addition to the network's accumulated learning.

When the music begins, the network trains several new states, usually within the first few seconds. The time interval between retraining then tends to increase, depending upon the character of the improvisation and the consequent variance of S_{audio} over time. Retraining might be thought of as adaptation, sensitive to the conditions of the sonic environment. As the performance develops, new analysis states will approximate one or, more often, several of those previously obtained. The network is continually queried to evaluate how far the current state S_{audio} approximates any of those previously learned. For example, if four states have been previously learned, an output response of {0. 1. 0. 0.} would indicate certain recognition of state 2; {0.3 0.7. 0. 0.} would indicate that relative characteristics of states 1 and 2 are evidenced.

One limitation of real-time use of the network is that it is "off-line" for this recognition when training is underway. It cannot report on current behaviour and map this assessment onwards. The time-period necessary to obtain an acceptable error during training increases dramatically as the number of output notes increases. This imposes a practical limit of c. 20 output nodes, which results in a maximum of c. 45 seconds for real-time training.

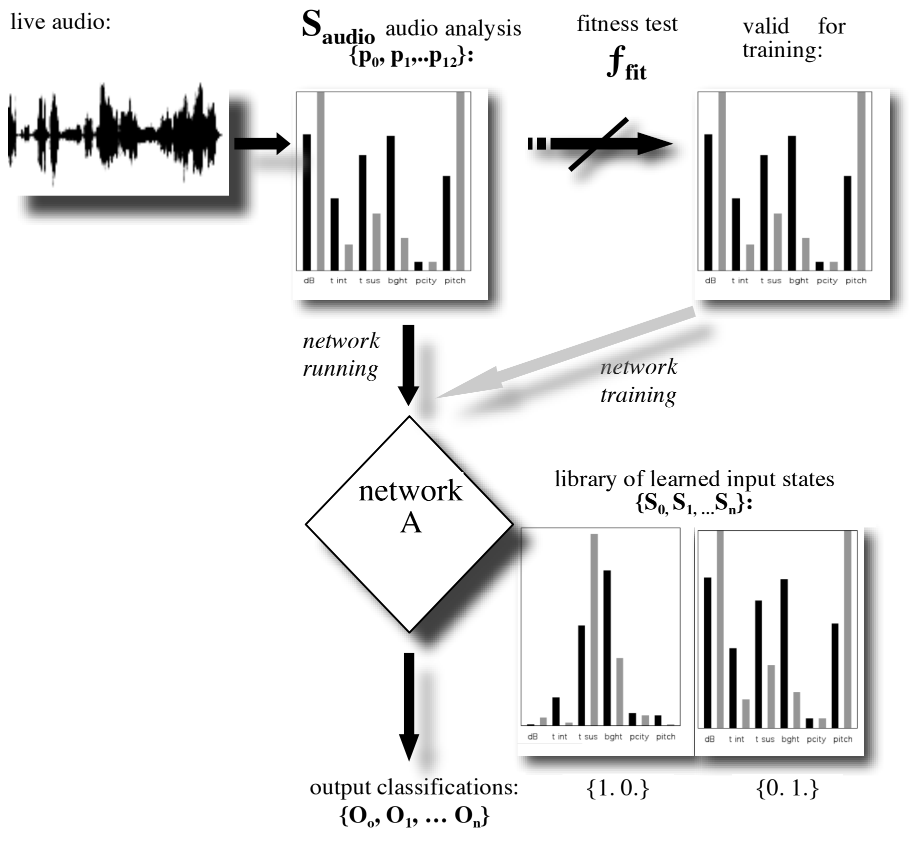

Fig. 2. Audio analysis and training of network A for classification.

2.3 Maps and Synthesis: $f(h) \rightarrow Q$.

Network mapping and synthesis is shown in figure 3 below. A second network (B) is deployed, trained in advance to generate synthesis functions Q, in response to 'ideal' (i.e. very simple) input conditions. The number and meaning of the resultant synthesis parameters is specific to each instance of the system: MIDI data for *au(or)a* and various sample playback and modification data for *piano_* and *cello_prosthesis*.

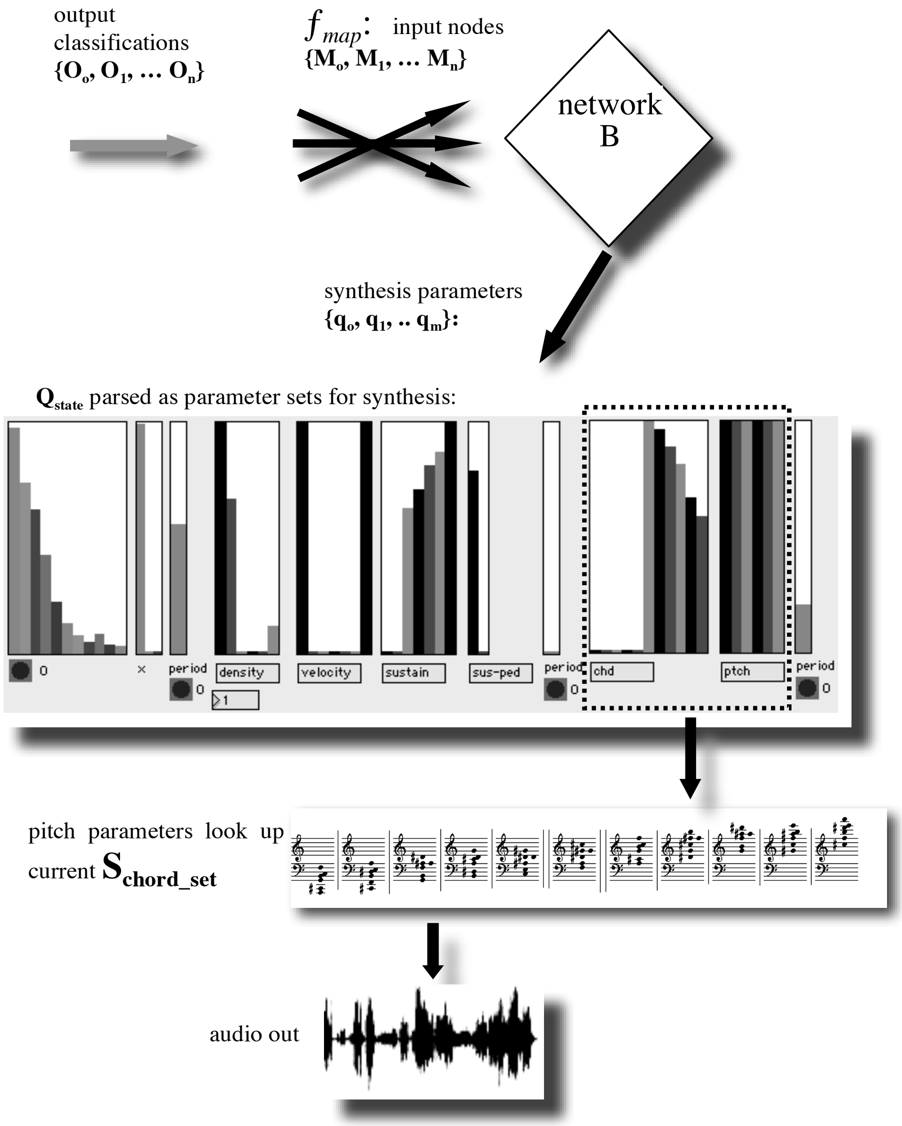

Fig. 3. Mapping to Network B to create parameters for stochastic synthesis.

A second independent network offers several advantages. Firstly, it provides transparency in the classification processes (which would otherwise be embedded within a single network).

More significantly, it allows for **covert mapping** between networks. The expanding list of outputs (i.e. classifications) from network A , $\{O_0, O_1, \ldots O_n\}$ is mapped via the function f_{map}, which randomly re-sorts the indices of the data. This jumbling up of output and input nodes provides genuine opacity; it is covert, challenging the player to adapt as the system's behaviour diversifies. The player is invited to attempt to learn which performance actions elicit a given response, and even if this is not a simple or attainable task, the process itself is closely related to the experience of human-only group improvisations.

Network B creates new input nodes as the list $\{M_0, M_1, \ldots M_n\}$ increases, which in turn allows the network to access more data from its previously learned set of outputs; this library of potential outputs constitutes the 'knowledge-base' of the system. It is decisive in characterising the music; a framework, a field of relations for aesthetic judgement.

Lastly, network A outputs are mapped with a power function to expand the classification set, i.e. to converge on the highest result. This becomes more apposite as the number of classifications increases. Consequently, network B is more likely to produce an output with a well-defined profile (as opposed to a more equally-weighted, and, amorphous, composite) even if this represents more than one original defined state.

Sound synthesis. Sound events are generated stochastically, in a method tolerant to the contingencies of the neural network output and complementary to the statistical method used for analysis. Stochastic techniques are well established in notated music and synthesis [20]; for NN Music, highly complex, mutable musical behaviours can be generated from an evolving probability distribution (or 'parameter profile') that is a composite of well-defined, theoretical, network outputs. As a consequence of this approach, and depending on the rate of iteration, the sonic environment can develop a "laminal" (textural) character or be more definitively "atomized" (event-based); codifications of sound established in free improvisation [21]. In the case of *_prosthesis*, sounds may be similar or timbrally distinct to the live instrument due to electronic transformation.

The behaviour of network B is entirely dependent on the classifications made by network A as it runs. If a player suggests three previously learned performance states, this will be reflected in a composite of three output synthesis states, summed in proportion to the network A output classification.

The final output of network B is Q_{state}; a list parsed into subsets according to parameter type. In figure 3, values for Q_{state} are shown at a given moment, each subset shown as a separate table. (Normally, due to the varying outputs of network B as it runs, Q_{state} is constantly changing). Q_{state} is then accessed as a probability distribution; each time a sound event is triggered, all subsets are invoked to determine the various characteristics and modifications of the event. The values indicated by the y axis in each subset denote the relative probability of a particular x axis value to be selected. Consequently, the network does not directly determine events, but constantly reshapes the stochastic distribution of their characteristics.

For example, musical timing is determined by the first three parameter sets These three processes aim to provide a sophisticated rhythmic vocabulary and structural syntax akin to those available to an improviser:

- A geometrically expanding series of 11 values: 53ms – 53^{11}ms
- The probability of selecting any one of three values for *i* (stretch factor).
- The probability that timing will stabilise into a periodic rhythmic pattern. The most recent 11 durations are recorded; for every new iteration there is a probability that these values – or a selected number of them – will be recalled rather than fresh values generated, creating a looped rhythm.

Pitches are determined by two parameter subsets, which are cross-referenced to the independent $S_{chord\text{-}set}$ corpus:

- The hexachords from $S_{chord\text{-}set}$ available for use.
- The note position allowed within each hexachord (1 to 6).

The outcome of the hexachord/note position is then referred to the current $S_{chord\text{-}set}$, from which the actual pitch is obtained. These techniques are extended to include a range of MIDI data for *aur(or)a,* and sample playback/transformation data, such as filtering, ring modulation and granular synthesis in *piano_prosthesis* and *cello_prosthesis*. The Q_{state} function can easily be generalized for any desired synthesis technique appropriate to the iterative method used.

3. Conclusion

The NN Music system comprises a web of analysis and synthesis functions, linked by a number of functional mapping and hidden algorithms, including the principle methods of unsupervised learning and classification on-the-fly, and covert parameter mapping. The modular approach follows the proposed PQ*f* model for improvisation systems, which indicates how individual components may be replaced, generalised or enhanced without undermining the structure of the whole. The system evidences, to some extent, attributes of a 'live algorithm': adaptability, empowerment, intimacy and opacity – aspiring to unimagined outcomes.

Future developments will need to address the time-delay problem incurred by on-the-fly training, and the consequent practical limit on the number of output nodes (analysis classifications). Other algorithms, such as k-means clustering, may offer more efficient methods for classification. The fitness function, which intercedes in network training should ideally be adaptive or unsupervised if the system is autonomous and entirely 'empowered'. Recurrent neural networks may offer new possibilities in bringing together adaptive and creative generative processes. In addition, greater insights into the improviser's performance, at appropriate structural levels would provide better material for network training, and impact on the responses of the system as a whole

The ultimate aim is to provide a stimulating and challenging environment for improvisers, which examines the liminal space between composition (intentional designs) and improvisation (collaborative or intuitive actions) in a musically convincing way. Artificial intelligence and learning offer great potential for further creative exploration of this.

Acknowledgements. The performers who have worked enthusiastically with the system and helped its development: Kate Ryder, Roger Redgate, Neil Heyde, to Goldsmiths Electronic Music Studios, and to Olivier Pasquet for *op.fann.mlp*.

References

1. Blackwell, T., Young, M.: Live Algorithms. Artificial Intelligence and Simulation of Behaviour Quarterly. 122, 7--9 (2005)
2. Young, M.: NN Music: Improvising with a 'Living' Computer. In: Proc.of the International Computer Music Conference. ICMA, San Francisco (2007)
3. Young, M.: Au(or)a: Exploring Attributes of a Live Algorithm. Electroacoustic Music Studies Network Conference. (2007). http://www.ems-network.org/spip.php?rubrique49
4. Lewis, G. E.: Too Many Notes: Computers, Complexity and Culture in Voyager. Leonardo Music Journal. 10, 33–-39, MIT Press, (2000)
5. Miranda E.R, Biles J.A.: Evolutionary Computer Music. Springer-Verlag: London. (2007).
6. Blackwell, T., Young, M.: Self-Organised Music. Organised Sound. 9:2, 123--136. Cambridge University Press. (2004)
7. Bastien, B. T. and Hostager, T.: Cooperative as communicative accomplishment: a symbolic interaction analysis of an improvised jazz concert. Communication Studies, 43, 92--104. (1992)
8. Rao, A. S., Georgeff, M. P.: Modeling rational agents within a BDI-architecture. In: 2nd Internationl Conference. on the Principles of Knowledge Representation and Reasoning. 473--484. Morgan Kaufmann, CA. (1991)
9. Hermann T., Ritter H.: Sound and Meaning in Auditory Data Display. IEEE Special Issue on Engineering and Music - Supervisory Control and Auditory Communication. 92:4, 730--741 (2004)
10. Eco. U.: The Open Work. Trans. Anna Cancogni. Harvard University Press. (1989).
11. Wessel, D. and Wright, M. Problems and Prospects for Intimate Musical Control of Computers. Computer Music Journal. 26:3. 11—22. (2002)
12. Csikszentmihalyi, M.: Flow: The Psychology of Optimal Experience. Harper Collins. (1991)
13. Sawyer, R. K.: Group creativity: Music, Theater, Collaboration. Lawrence Erlbaum Associates. (2003)
14. Boulez, P.: Sonate, que me veux-tu? In: Orientations: Collected Writings. London: Faber and Faber. (1986)
15. Adorno, T.:Vers une Musique Informelle. In: Quasi une Fantasia, trans. Rodney Livinstone. London: Verso. (1963)
16. Stojanov, G., Stojanoski, K.: Computer Interfaces: From Communication to Mind-Prosthesis Metaphor. In: Cognitive Technology: Instruments of Mind. LNCS, vol. 2117. pp. 301 --311. Springer, Heidelberg (2001)
17. Toivianen, P.: Symbolic AI versus Connectionism in Music Research. In: Miranda, E. (ed.) Readings in Music and Artificial Intelligence. Harwood Academic (2000)
18. Koblyakov, L.: Pierre Boulez: A World of Harmony. Harwood Academic (1990).
19. Bailey, D.: Improvisation: Its Nature and Practice in Music. Da Capo Press (1992)

20. Xenakis, I.: Formalized Music: Thought and Mathematics in Composition. Rev. Ed. Pendragon Press (2001)
21. Prevost, E.: No Sound Is Innocent: AMM and the Practice of Self-invention. Copula (1995)