Anticipatory Model of Musical Style Imitation using Collaborative and Competitive Reinforcement Learning

Arshia Cont, Shlomo Dubnov, Gerard Assayag

Anticipatory Model of Musical Style Imitation using Collaborative and Competitive Reinforcement Learning

Arshia Cont[1,2], Shlomo Dubnov[2], and Gérard Assayag[1]

[1] Ircam - Centre Pompidou - UMR CNRS 9912, Paris.
{cont,assayag}@ircam.fr
[2] Center for Research in Computing and the Arts, UCSD.
sdubnov@ucsd.edu

Abstract. The role of *expectation* in listening and composing music has drawn much attention in music cognition since about half a century ago. In this paper, we provide a first attempt to model some aspects of musical expectation specifically pertained to short-time and working memories, in an anticipatory framework. In our proposition *Anticipation* is the mental realization of possible predicted actions and their effect on the perception of the world at an instant in time. We demonstrate the model in applications to automatic improvisation and style imitation. The proposed model, based on cognitive foundations of musical expectation, is an active model using reinforcement learning techniques with multiple agents that learn competitively and in collaboration. We show that compared to similar models, this anticipatory framework needs little training data and demonstrate complex musical behavior such as long-term planning and formal shapes as a result of the anticipatory architecture. We provide sample results and discuss further research.

1 Introduction

About half a century ago, the musicologist Leonard Meyer drew attention to the importance of *expectation* in the listener's experience of music. He argued that the principal emotional content of music arises through the composer's choreographing of expectation [1]. Despite this significance, musical expectation has not enjoyed its cognitive importance in existing computational systems which mostly favor prediction-driven architectures without enough cognitive constraints. In this paper, we will introduce a first attempt towards modeling musical systems with regards to the psychology of musical expectations. For modeling these constraints, we use anticipatory systems where several accounts of musical expectation is modeled explicitly. We claim that such cognitive modeling of music would constitute complex musical behavior such as long-term planning and generation of learned formal shapes. Moreover, we will show that the anticipatory approach greatly reduces the dimensions of learning and allows satisfactory performance when little data is available.

We start the paper by studying the cognitive foundations of music as the core inspiration for the proposed architecture. In section 2, we discuss important aspects of the psychology of music expectation such as auditory learning, mental representations of expectations and auditory memory. Our hope is that such studies create motivations for modeling complex musical behavior as proposed.

We demonstrate our system in applications to automatic music improvisation and style imitation as a direct way to showcase complex musical behavior. Musical style modeling consists of building a computational representation of the musical data that captures important stylistic features. Considering symbolic representations of musical data, such as musical notes, we are interested in constructing a mathematical model such as a set of stochastic rules, that would allow creation of new musical improvisations by means of intuitive interaction between human musicians or music scores and a machine. In section 3 we study some of the important approaches in the literature for the given problem. We will be looking at these systems from two perspectives: that of representation and memory underlying the challenge of dimensionality of music information, and that of modeling addressing learning and grasping stylistic features.

Section 4 provides background on anticipatory modeling used in the proposed system. It also contains the main idea behind our anticipatory modeling of musical expectation. Our design explicitly addresses two types of anticipatory models introduced in [2]: state anticipation and payoff anticipation. In this work, we tend to model two aspects of musical expectations, namely dynamic adaptive and conscious expectations as discussed in section 2.

The proposed architecture features reinforcement learning as an interactive module between the system and an outside environment and addresses adaptive behavior in auditory learning. In general, our model is a modular system that consists of three main modules: *memory, guides* and *learning*. The memory module serves for compact representation and future access of music data. Guides are reinforcement signals from the environment to the system or from previous instances of the system onto itself that guide the learning module to relevant places places in memory for updates and learning. The learning module captures stochastic behavior and planning through interactive learning. In section 5 we overview the general architecture and each module will be then presented separately in sections 6 to 8. After the design, we demonstrate some generation results in section 9. Results show evidence of long-term planning achieved through learning as an outcome of anticipatory modeling of working memory in music cognition. Furthermore, the system requires much less training data compared to similar systems, again due to the proposed anticipatory framework. We end this chapter by discussing the complexity of the proposed anticipatory architecture and future works.

2 Cognitive Foundations

The core foundations of the proposed model in this chapter is based on the *psychology of musical expectation*. In his recent book, David Huron vastly studies

various aspects of music expectation [3]. Here, we highlight important aspects of his work along other cognitive facts pertinent to our proposal.

2.1 Auditory Learning

There are extensive evidence for the *learning* aspect of musical expectation through auditory learning and in opposition to innate aspects of these behaviors. One of the most important discoveries in auditory learning, has been that listeners are sensitive to the probabilities of different sound events and patterns, and these probabilities are used to form expectations about the future. An important research landmark in favor of this claim is the work in [4]. On the other hand, brain does not store sounds. Instead, it interprets, distills and represent sounds. It is suggested that brain uses a combination of several underlying presentations for musical attributes [3]. A good mental representation would be one that captures or approximates some useful organizational property of an animal's actual environment.

But how does the brain know which representation to use? Huron suggests that expectation plays a major role. There is good evidence for a system of rewards and punishments that evaluates the accuracy of our unconscious predictions about the world. Our mental representations are being perpetually tested by their ability to usefully predict ensuing events, suggesting that *competing and concurrent representations* may be the norm in mental functioning [3]. This view is strongly supported by the neural Darwinism theory of Edelman [5]. According to this theory, representations compete with each other according to Darwinian principles applied to neural selection. Such neural competition is possible only if more than one representation exists in the brain. In treating different representations and their expectation, each listener will have a distinctive listening history in which some representations have proved more successful than others.

2.2 Mental Representations of Expectation

According to Huron, memory does not serve for recall but for *preparation*. In chapter 12 of his book, Huron tries to address the structure rather than content of mental representations and introduces a taxonomy for auditory memory that constitutes at least four sources of musical expectations as follows:

Veridical Expectation: Episodic Memory is an explicit memory and a sort of autobiographical memory that holds specific historical events from our past. Episodic memory is easily distorted and in fact, the distortion occurs through repeated retelling or recollection. Most importantly, our memories for familiar musical works are episodic memories that have lost most of their autobiographical history while retaining their accuracy. This sense of familiarity or expectation of familiar works is refereed to, by Huron and Bharucha, as *Veridical expectation.*

Schematic Expectation: Schematic expectation is associated with *Semantic memory*; another type of explicit memory which holds only declarative

knowledge and is distinguished from episodic by the fact that it does not associate the knowledge to any historical past but as stand-alone knowledge. This kind of memory is most active in first-exposure listening (when we do not know the piece) where our past observations and learned schemas are generalized. These sort of auditory generalizations are reminiscent of the learned categories characteristic of semantic memory.

Dynamic Adaptive Expectation: Expectation associated with Short-term memory is *Dynamic Adaptive Expectation*. It occurs when events do not conform with expectations that have been formed in the course of listening to the work itself. These expectations are updated in realtime especially during exposure to a novel auditory experience such as hearing a musical work for the first time.

Conscious Expectation: All the three types of expectations discussed above are unconscious in origin. Another important class of expectations arise from conscious reflection and prediction. Such explicit knowledge might come from external sources of information (such as program notes) or as part of a listener's musical expertise, or even arise dynamically while listening to a novel musical work. An argument for the last type, and most important for this work, is the perception of musical form during listening. This form of expectation come from the mental desktop psychologists refer to as working memory.

All these expectation schemes operate concurrently and in parallel. Schematic expectations are omnipresent in all of our listening experiences. When listening to a familiar work, the dynamic-adaptive system remains at work – eventhough the veridical expectation anticipates exactly what to expect. Similarly, when listening for the first time to an unfamiliar work, the veridical system is constantly searching for a match with familiar works. The veridical system is essential for catching the rare moments of musical quotation or allusion. In short, an anticipatory effect such as *surprise* is a result of various types of interactions among these lower-level components of music expectation cognition. For a thorough discussion see [3].

An ideal anticipatory model of music cognition should address all four types of expectations addressed above. However, for this work as a first attempt, we focus on *dynamic adaptive expectation* and *conscious expectation* and address the rest in future works. For the latter, we are interested in the conscious expectations that arise dynamically while listening to a "new" musical work.

2.3 Memory and Reinforcement

The role of memory in the brain for music might hint us on how musical representations are stored and how they interact within themselves in the brain and with an outside environment. In the previous section we looked at the representational aspect of memory with regard to music expectation and here briefly introduce the interactive level. This interactive level should guide us on how we can model memory access and learning for our purpose.

Snyder in [6] proposes an auditory model of memory that consists of several stages, from which we consider feature extraction, Long Term Memory (LTM) and Short Term Memory (STM). Feature extraction is some sort of perceptual categorization and grouping of data. Events processed at this stage can activate those parts of LTM evoked by similar events in the past. Activated LTM at this point form a context for current awareness. This context takes the form of expectations that can influence the direction that current consciousness takes. Memory also acts like a filter determining which aspects of our environment we are aware of at a given time. LTM that reaches this higher state of activation can then persist as current STM. Information in STM might be repeated or rehearsed. This rehearsal greatly *reinforces* the chances that the information being circulated in STM will cause modifications in permanent LTM. We consider both activation and reinforcement processes in our design of guide and learning modules.

Besides this unconscious level of reinforcement, like sensory representations, conscious thinking also requires some guidance and feedback to ensure that thinking remains biologically adaptive [3]. Useful thinking needs to be rewarded and encouraged, while useless thinking needs to be suppressed or discouraged.

3 Background on Stochastic Music Modeling

In this section we look at several prior attempts to modeling music signals either for generation (automatic improvisation or style imitation) or modeling long-term dependencies observed in music time series. In this work, we are interested in *automatic* systems where there are no rules or a priori information abducted into the system by experts and everything is learned through the life-span of the system. Moreover, we are interested in systems which address directly the complexity of music signals as will be clear shortly.

Earlier works on style modeling employed information theoretical methods inspired by universal prediction. In many respects, these works build upon a long musical tradition of statistical modeling that began in the 50s with Hiller and Isaacson's "Illiac Suite" [7] and Xenakis using Markov chains and stochastic processes for musical composition [8]. In what follows, we will review some of the state-of-the-arts systems proposed in the literature from two standpoints: their musical representations and stochastic modeling.

3.1 Musical Representation

Music information has a natural componential and sequential structure. While sequential models have been extensively studied in the literature, componential or multiple-attribute models still remain a challenge due to complexity and explosion in the number of free parameters of the system. Therefore, a significant challenge faced with music signals arises from the need to simultaneously represent and process many attributes of music information. The ability (or inability)

of a system to handle this level of musical complexity can be revealed by studying its ways of musical representations or memory models both for storage and learning. Here, we will compare different memory models used and proposed in the literature for systems considering this complex aspect of music signals. We will undertake this comparison by analytically looking at each model's complexity and its modality of interaction across attributes which in terms, determine its power of (musical) expressivity. We will be looking at *cross-alphabets* [9–11], *multiple-viewpoints* [12] and mixed memory *Factorial Markov models* [13].

In order to better understand each model in this comparison, we use a toy example demonstrated in Figure 3.1 containing the first measure of J.S.Bach's *two-part invention No.* 5 (Book II). The music score in figure 3.1 is parsed between note onsets to obtain distinct events through time as demonstrated. In this article, we consider discrete MIDI signals as is clear from the figure. For the sake of simplicity, we only represent three most important attributes, namely pitch, harmonic interval and beat duration of each parsed event as shown in Table 3.1. This table represents 15 vector time series I_t corresponding to 15 parsed events in Figure 3.1 where each event has three components (i_t^μ). Let k denote the number of components for each vector I_t and n_ℓs denote the dictionary size for each attribute i_t^ℓ. Later in section 6, we will use the same example to demonstrate our alternative representation scheme.

Event Number I_t	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9	I_{10}	I_{11}	I_{12}	I_{13}	I_{14}	I_{15}
MIDI Pitch (i_t^1)	0	51	63	62	63	0	65	0	67	67	63	68	68	58	60
Harmonic Interval (i_t^2)	0	0	0	0	24	0	0	0	4	5	0	8	6	0	0
Duration (i_t^3)	1	1	1	1	1	1	1	2	1	1	1	1	1	1	1

Table 1. Music attributes for distinct events of parsed score in Figure 3.1

Cross alphabets The simplest model used so far in music applications is *cross-alphabet* where a symnol is a vector of multiple attributes. Therefore, cross-alphabet models are very cheap but they do not model interaction among components in any ways. To overcome this shortcoming, researchers have considered various membership functions to allow for these context dependencies through various heuristics [10, 11]. Such heuristics might make the system dependent upon the style of music being considered or reduce generalization capabilities. Moreover, as the number of components (or dimensions) increase this representation becomes less informative of the underlying structure.

In our toy example each symbol of the alphabet is a unique 3-dimensional vector. Note that in this specific example, there are 15 alphabets since none of them is being reused despite considerable amount of modal interactions among components and high autocorrelations of each independent component.

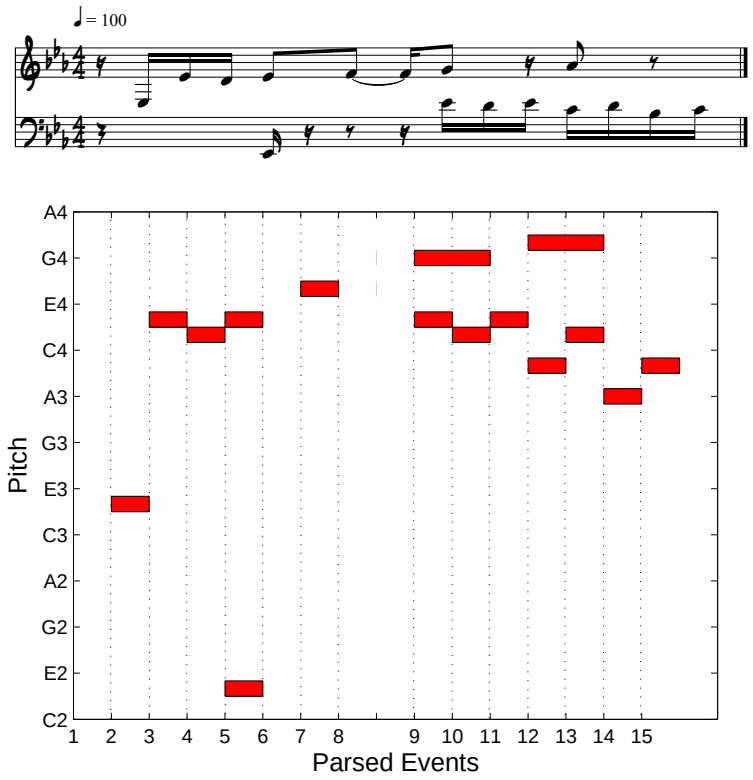


Fig. 1. Parsed pianoroll presentation for the first measure of J.S.Bach's *two-part Invention No.5* (Book II) with quantization of $\frac{1}{16}$ beats

Multiple viewpoints *Multiple viewpoints model* [12] is obtained by deriving individual expert models for each musical attribute and then combining the results obtained from each model. This means that in the multiple viewpoint model of the above example, three other rows for two-dimensional representations of `<pitch, harmonic interval>`, `<pitch, duration>`, etc. and one row of three-dimensional vectors are added to the representation. At this stage, the model's *context* is constructed.

Multiple viewpoint models are more expressive than cross-alphabet models since by combining models we allow modal interactions among components. Moreover, the system can reach parts of the hypothesis space that the individual models would not be able to reach. However, the context space is obviously too large and hence, learning requires huge repertoire of music for training data to generate few musical phrases [12]. In our toy example, with 9 distinct pitches, 6 distinct harmonic intervals and 2 durations, the state-space of this model amounts to $9 + 6 + 2 + 54 + 18 + 12 + 108 = 209$.

Factorial Markov Models Mixed Memory models are geared to situations where combinatorial structure of state space leads to an explosion of the number of free parameters. But unlike the above methods, the alphabets of the dictionary is assumed known instead of them being added online to the system. Factorial Markov models, model the coupling between components in a compact way.

To obtain a compact representation, we assume that components at each time t are conditionally independent given the previous vector event at $t-1$, or

$$P(I_t|I_{t-1}) = \prod_{\nu}^{k} P(i_t^{\nu}|I_{t-1}) \tag{1}$$

and that the conditional probabilities $P(i_t^{\nu}|I_{t-1})$ can be expressed as a weighted sum of "cross-transition" matrices:

$$P(i_t^{\nu}|I_{t-1}) = \sum_{\mu=1}^{k} \phi^{\nu}(\mu) a^{\nu\mu}(i_t^{\nu}|i_{t-1}^{\mu}) \tag{2}$$

where $\phi^{\nu}(\mu)$s are positive numbers that satisfy $\sum_{\mu} \phi^{\nu}(\mu) = 1$ and measure the amount of correlation between the different components of the time series. A non-zero $\phi^{\nu}(\mu)$ means that all the components at *one time step* influence the νth component at the next. The parameters $a^{\nu\mu}(i'|i)$ are $n \times n$ are transition matrices and provide a compact way to parameterize these influences [13].

The number of free parameters in eq. 2 is therefore upper-bounded by $O(k^2n^2)$ (where n denote $\max n_i$[3]) and the state-space size is $\prod_i n_i$. In our toy example the state-space size of the system would be $9 \times 6 \times 2 = 108$.

3.2 Stochastic Modeling

In this section, we review the systems mentioned above in terms of their ways of learning stochastic rules or dependencies from given musical sequences in order to generate new ones in the same style of music.

The most prevalent type of statistical model encountered for music are *predictive* models based on *context* implying general Markov models [14]. Universal prediction methods improved upon the limited memory capabilities of Markov models by creating context dictionaries from compression algorithms, specifically using the Lempel-Ziv incremental parsing [15] , and employing probability assignment according to Feder et al. [16]. Music improvisation was accomplished by performing a random walk on the phrase dictionary with appropriate probabilistic drawing among possible continuations [17, 9, 11]. Later experiments explored Probabilistic Suffix Tree (PST) [18], and more recently in [10] using Factor Oracle (FO) [19]. Other methods include the use of Genetic Algorithms [20] and neural networks [21] just to name a few.

[3] In original paper of Factorial Markov models, the authors assume that the dictionary sizes are all the same and equal to n. For the sake of comparison we drop this assumption but keep n as defined above to obtain the coarse definition in equation 2.

The inference and learning structures for Multiple Viewpoint Models (section 3.1) can be categorized as *Ensemble Learning* algorithms and have had multiple manifestations [22, 12]. One advantage of this type of modeling is the explicit consideration of long-term dependencies during learning where they combine the viewpoint predictions separately for long-term and short-term models [22]. Due to the explosion of parameters, results of learning are hard to visualize and assess. Their generation samples are usually few monophonic bars out of learning on an entire database of music (e.g. all Bach chorals).

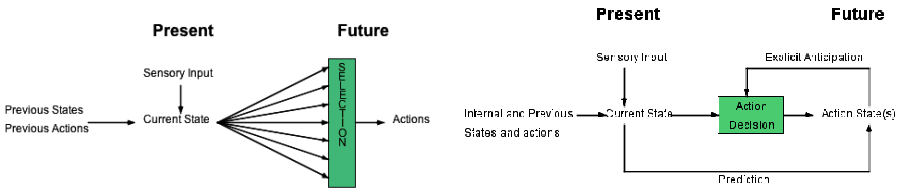
Despite the explicit componential representation of Factorial Markov Models, the correlation factors $\phi^{\nu}(\mu)$ model only *one step* dependencies and lack modeling long-term behavior, essential in computational models of music. Correspondingly, authors use this method to analyze correlations between different voices in componential music time series without considering generation [13].

Another main drawback of the above methods is lack of responsiveness to changes in musical situations that occur during performance, such as dependence of musical choices on *musical form* or changes in *interaction* between players during improvisation. Interaction has been addressed previously in [11] for PST based improvisation by means of a fitness function that influenced prediction probabilities according to an ongoing musical context, with no consideration of planning or adaptive behavior. Statistical approaches seem to capture only part of the information needed for computer improvisation, i.e. successfully modeling a relatively short term stylistics of the musical *surface*. Although variable Markov length and universal methods improve upon the finite length Markov approach, they are still insufficient for modeling the true complexity of music improvisation.

4 Background on Anticipatory Modeling

All of the systems reviewed in the previous section are based on predictions out of a learned context. In this work, we extend this view by considering *musical anticipation* and in accord with the psychology of musical expectation. Anticipation is different from both prediction and expectation. An anticipatory system, in short, is "a system containing a *predictive model* of itself and of its *environment*, which allows it to change state at an instant in accord with the model's predictions pertaining to a later instant" [23]. More concretely, *Anticipation* is the mental realization of possible predicted actions and their effect on the perception and learning of the world at an instant in time. Hence, anticipation can be regarded as a marriage of actions and expectations. In this framework, an anticipatory system is in constant interaction with an outside environment for which, it possesses an internal predictive model. In an anticipatory system, action decisions are based on future predictions as well as past inference. It simulates adaptive frameworks in the light of different behaviors occurring in interaction between the system with itself and/or its environment. In this view, the anticipatory effect can be described as a reinforcing feedback as a result of the interaction between the system and the environment onto the system.

In [2] Butz et al. draw distinction between four types of anticipation for modeling: *Implicit, Payoff, Sensory*, and *State* anticipations. We did not find a direct correspondence between those mentioned in section 2. The proposed system in this chapter is both a *payoff anticipatory system* and *state anticipation system*. Figure 2 shows the diagrams for both models separately and how they use future predictions for decision making. The system proposed hereafter is state anticipatory because of the explicit use of prediction and anticipation during both learning and decision making. It is also a payoff anticipatory system because of the selective behavior caused by the collaborative and competitive learning and generation discussed in section 8. From a musical standpoint following our introduction in section 2, we attempt implicit modeling of short-term and working memories responsible for dynamic adaptive expectation and long-term planning.



(a) Payoff Anticipation (b) State Anticipation

Fig. 2. Anticipatory Modeling diagrams used in the proposed system.

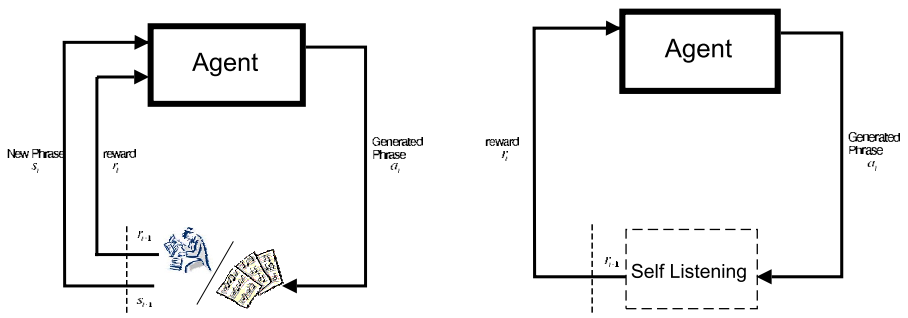
Davidsson in [24] proposes a framework for preventive anticipation where he incorporates collaborative and competitive multiple agents in the architecture. While this has common goals with our proposal, it is different since Davidsson uses rule-based learning with ad-hoc schemes for collaboration and competition between agents. Recently, in the computer music literature, Dubnov has introduced an anticipatory information rate measure that when run on non-stationary and time varying data such as audio, can capture anticipatory profile and emotional force data that has been collected using experiments with humans [25].

5 General Architecture

After the above introduction, it is natural to consider a *reinforcement learning (RL)* architecture for our anticipatory framework. The reinforcement learning problem is meant to be a straightforward framing of the problem of learning from interaction to achieve a goal. The learner and decision-maker is called the agent. The thing it interacts with, comprising everything outside the agent, is called the environment. These interact continually, the agent selecting actions and the environment responding to those actions and presenting new situations to the agent. The environment also gives rise to rewards, special numerical values

that the agent tries to maximize over time. This way, the model or agent is interactive in the sense that the model can change through time according to reinforcement signals sent by its environment. Any RL problem can be reduced to three signals passing back and forth between an agent and its environment: one signal to represent the choices made by the agent (the actions), one signal to represent the basis on which the choices are made (the states), and one signal to define the agent's goal (the rewards)[26]. In a regular RL system, rewards are defined for goal-oriented interaction. In musical applications, defining a *goal* would be either impossible or would limit the utility of the system to a certain style. In this sense, the rewards used in our interaction are rather *guides* to evoke or repress parts of the learned model in the memory, as discussed in section 7.

In our system, agents are learners and improvisors based on a model-based RL *dyna* architecture [26]. Here, the environment is anything that lies outside the agent, or in this case, a human performer or a music score fed sequentially into the system. Each agent has an internal model of the environment and adapts itself based on new musical phrases and rewards it receives at each interaction. For our purpose, we propose two execution modes as demonstrated in Figure 3. In the first, referred to as *Interaction mode*, the system is interacting either with a human performer (live) for machine improvisation or with music score(s) for style imitation and occurs when external information is being passed to the system from the environment. During the second mode, called *self listening mode*, the system is in the generation phase and is interacting with itself.



(a) Interaction Mode (b) Self Listening Mode

Fig. 3. Machine Improvisation modes diagram

The internal models in agents play the role of *memory* and *mental representations* of input sequences from the environment and will be detailed in the following section. At each instance of interaction, the agents update their models and learn strategies as discussed in section 7 and using guides or rewards presented in section 8.

6 Musical Representation

Representation of musical sequences in our system serves as musical memory, mental representation of music signals and internal models of the agents. A single music signal has multiple attributes and as stated earlier, each attribute is responsible for an individual mental representation which *collaborates* and *competes* with others during actions and decision making. This collaboration and competition is handled during learning and is discussed in section 8. For now, it suffices to say that the agent in both modes of interaction in Figure 3 consists of multiple agents, each responsible for *one* musical attribute. This feature is of great importance since it reduces the dimensionality of the system during learning, allowing it to interact when small data is available and in a fast way. The number of attributes and nature of them are independent of the agent architecture. For this experiment, we hand-picked 3 different attributes (pitch, harmonic interval and quantized duration in beats) along with their first difference, hence a total of 6. Upon the arrival of a MIDI sequence, it is quantized, cut into polyphonic "slices" at note onset/offset positions, and then different viewpoints are calculated for each slice. Slice durations are represented as multiples of the smallest significant time interval that a musical event would occupy during a piece (referred to as *tatum*). For demonstration, table 6 shows these features as time series data calculated over the score of figure 3.1.

Event Number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Pitch (MIDI)	0	51	63	62	63	0	65	0	67	67	63	68	68	58	60
Harmonic Int.	0	0	0	0	24	0	0	0	4	5	0	8	6	0	0
Duration	1	1	1	1	1	1	1	2	1	1	1	1	1	1	1
Pitch Diff..	0	0	12	-1	1	0	0	0	0	0	1	-3	0	-4	2
Harm. Diff.	0	0	0	0	0	0	0	0	0	1	0	0	-2	0	0
Dur. Ratio	1	1	1	1	1	1	1	2	0.5	1	1	1	1	1	1

Table 2. Time series data on the score of figure 3.1 showing features used in this experiment

After the data for each viewpoint is gathered it has to be represented and stored in the system in a way to reflect principles discussed in section 2. Of most importance for us are the expressivity of the model, least computational complexity and easy access throughout the memory model. There are many possible solutions for this choice. In our multiple-agent framework, we have chosen to store each attribute as a *Factor Oracle (FO)* [19]. In this paper, we give a short description of the properties and construction of FO and leave the formal definitions and musical interests in [19, 10]. Basically a factor oracle is a finite state automaton learned incrementally in linear time and space. A learned sequence of symbols $A = a_1 a_2 \cdots a_n$ ends up in an automaton whose states are $s_0, s_1, s_2 \cdots s_n$. There is always a transition arrow labeled by symbol a_i going from state s_{i-1} to state s_i. Depending on the structure of A, other arrows may appear: *forward transi-*

tions from a state s_i to a state s_j, $0 \leq i < j <= n$, labeled by symbol a_j; *suffix links*, directed backward and bearing no label. The forward transitions model a factor automaton, that is every factor $p = a_i a_{i+1} \cdots a_{j-1} a_j$, $1 \leq i \leq j \leq n$ in A corresponds to a unique transition path labeled by p, starting in s_0 and ending in state s_j. Suffix links connect repeated patterns of A, i.e. states sharing large common suffixes. In general, given a sequence, the constructed FO returns two *deterministic* functions: a transition function $F_{trn} : S \times \Sigma \rightarrow \{S \cup \emptyset\}$ and suffix links $F_{sfx} : S \rightarrow \{S \cup \emptyset\}$, where S is the set of states and Σ is the alphabet on which A is constructed. Figure 4 shows four instances of FO construction over data presented in table 6.

An important property of FO for this work is their power of generation. Navigating the oracle and starting in any place, following forward transitions generates a sequence of labeling symbols that are repetitions of portions of the learned sequence; following one suffix link followed by a forward transition generates an alternative path in the sequence, creating a recombination based on a shared suffix between the current state and the state pointed at by the suffix link. This shared suffix link is called context in context-inference models. In addition to completeness and incremental behavior of this model, the best suffix is known at the minimal cost of just following one pointer. By following more than one suffix link before a forward jump or by reducing the number of successive factor link steps, we make the generated variant less resemblant to the original.

7 Environmental Interactions

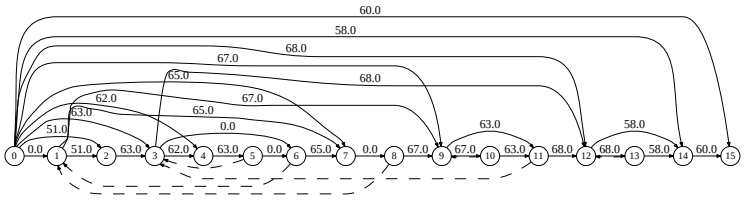
Guide signals received from the environment are an essential part of the proposed system since they define the sensitivity of the system to outside world, directions it can take and its musical capabilities. When at time t the new music sequence $A^t = a_1 a_2 \cdots a_N$ is received from the environment, an ideal reward signal should reinforce parts of memory which most likely evoke the sequence received to be able to generate recombinations or musically meaningful sequences thereafter. In the RL framework, this means that we want to assign numerical rewards to *transition states* and *suffix states* of an existing Factor Oracle with internal states s_i. Guide computation occurs using the previously learned FOs (defined by FO^{t-1}) and before incorporating the new sequence into the model.

After different attributes of A^t are extracted as separate sequences each in form $\{x_1 \ldots x_N\}$, we use a *probability assignment function* P from $S^* \rightarrow [0, 1]$ (where S^* is the set of all n-tuples of states available to FO) to assign rewards to states in the model as follows:

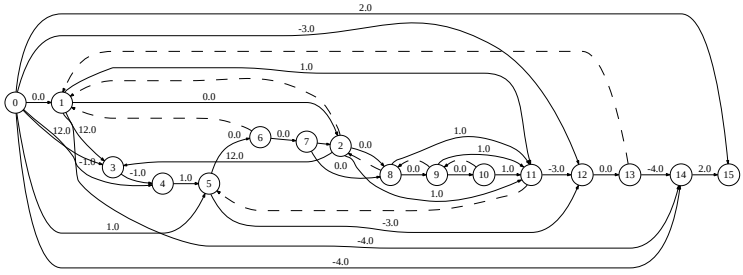
$$P(s_{1^*} s_{2^*} \ldots s_{N^*} | FO^{t-1}) = \left[\sum_{i=1}^{N} p(x_i | s_{i^*}) \right] / N \tag{3}$$

where

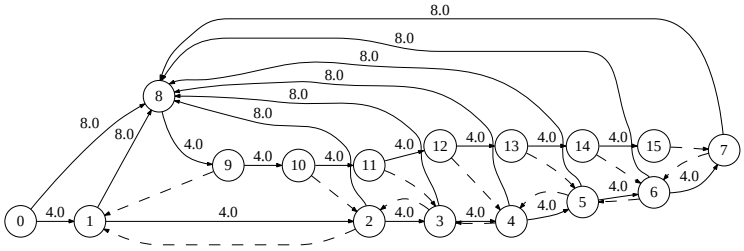
$$p(x_i | s_i^*) = \begin{cases} 1 & \text{if } F_{trn}(s_{i-1}^*, x_i) = s_i^* \\ 0 & \text{if } F_{trn}(s_{i-1}^*, x_i) = \emptyset \end{cases} \tag{4}$$



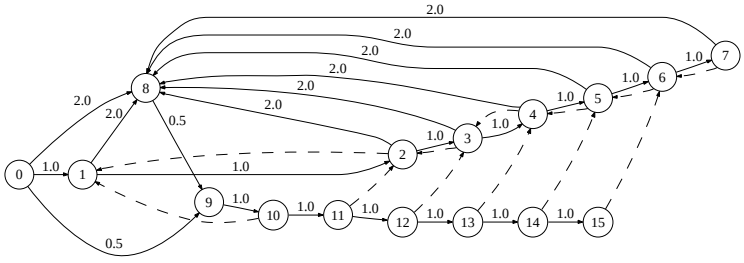
(a) Pitch FO



(b) Pitch Contour FO



(c) Duration FO



(d) Duration Ratio FO

Fig. 4. Learned Factor Oracles over pitch and duration sequences in table of table 6. Each node represents a state, each solid line a *transition* and dashed line a *suffix link*.

Of course the exploration of the search space S^* is optimized by not considering all possible n-tuples. Instead a simple forward checking strategy is used to reduce S^* to significantly rewarded subsets.

Rewards out of equation 3 reinforce the states in the memory that are factors of the new sequence A^t. In other words, it will *guide* the learning described later to the places in the memory that should be mostly regarded during learning and generation. For example, the reward for $\{s_{1^*} s_{2^*} \dots s_{N^*}\}$ would be equal to 1 if the state/transition path $s_{1^*} \dots s_{N^*}$ regenerate literally the sequence A^t.

To assign rewards to suffix links, we recall that they refer to previous states with largest common suffix. Using this knowledge, a natural reward for a suffix link would be proportional to the *length* of the common suffix that the link is referring to. Fortunately, using a factor oracle structure, this measure can be easily calculated online and has been introduced in [27]. Note that the process defined above assigns numerical values to states pertaining to associate paths (transitions or suffix links) in each FO. This value is an *immediate* reward, noted by $r(s_t, a_t)$ for emission of a symbol a_t while at state s_t.

Rewards or guides are calculated the same way for both modes of the system described before with an important difference. We argue that the rewards for the *interaction mode* (Figure 3(a)) correspond to a *psychological attention* towards appropriate parts of the memory and guides for the *self-listening mode* (Figure 3(b)) correspond to a *preventive* anticipation scheme. This means that while interacting with a live musician or sequential score, the system needs to be attentive to input sequences and during self-listening it needs to be preventive so that it would not generate the same path over and over. Moreover, these schemes provide the conscious reinforcement required to encourage or discourage useful and useless thinking as mentioned in section 2.3. This is achieved by treating environmental rewards with positive and negative signs appropriately.

8 Interactive Learning

Reinforcement Learning techniques are mostly studied within Markov Decision Process (MDP) framework. An MDP in general is defined by a set of states-action pairs $S \times A$, a reward function $R : S \times A \rightarrow \mathbb{R}$ and a state transition function $T : S \times A \rightarrow S$. Given this MDP, RL techniques aim to find the policy as a mapping probability $Q(s, a)$. To conform the representational scheme presented before to this framework, we define MDP state-action pairs as FO states and emitted symbol from that state. The transition function would then be the deterministic FO transition functions as defined before. This way the policy can be represented as a matrix Q which stores values for each possible state-action pair in a given FO.

In a regular reinforcement learning session, the system simulates itself up to a fixed number of episodes with terminal states, in order to maximize the overall reward during each episode by learning a Q matrix. At each interaction cycle with their environments, depending on their mode of interaction (Figure 3), the agents receive guides, update their existing models and learn new ones (as

FOs and only during the *interaction mode*), and learn policies through some *Q-learning* algorithm by simulating episodes of system run. In this view, one can say that during each learning episode the system is *practicing* or improvising fixed length pieces using what it has learned so far in order to adapt itself to new musical situations defined by the newly arrived sequence and to learn and update policies. The main cycles of the interactive learning is shown in algorithm 1. This architecture is based on Dyna [26] with multiple agents and FOs as models. This cycle happens at each interaction between the system and the environment. During the self-listening mode, the algorithm is the same except that FOs are not updated.

1 Receive the new sequence A^t from the environment;
2 Compute guides on FO^{t-1}s;
3 Update FO^{t-1}s to FO^ts using A^t;
4 Learn policies (Q matrices);

Algorithm 1: Interactive Learning

Hereafter, we focus on the policy learning algorithm. At this stage, the algorithm simulates episodes of improvisations using previously learned policies and updates the Q matrices in order to maximize the environmental rewards. This RL module must conform to cognitive foundations presented in section 2, i.e. agents should be collaborative, competitive, and memory-based.

8.1 Competitive and Collaborative learning

As discussed in section 2.1, different mental representations of music work in a collaborative and competitive manner based on their predictive power to make decisions. This can be seen as kind of a *model selection* where learning uses all the agents' policies available and chooses the best one for each episode. This winning policy would then become the *behavior policy* with its policy followed during that episode and other agents being influenced by the actions and environmental reactions from and to that agent.

At the beginning of each episode, the system selects one agent using the probability in Equation 5, with positive parameter β_{sel}, and M as the total number of agents or attributes. Low β_{sel} causes equi-probable selection of all modules and vice versa. This way, a behavior policy π^{beh} is selected *competitively* at the beginning of each episode based on the value of the initial state s_0 among all policies π^i as demonstrated in Equation 5.

$$Pr(i|s_0) = \frac{e^{\beta_{sel} \sum_k Q^i(s_0,a_k)}}{\sum_{j=1}^{M} e^{\beta_{sel} \sum_r Q^j(s_0,a_r)}} \quad , \quad \pi^{beh} = \underset{i}{\operatorname{argmax}}\, Pr(i|s_0) \tag{5}$$

During each learning episode, the agents would be following the behavior policy. For update of π^{beh} itself, we can use a simple Q-learning algorithm but

in order to learn other policies π^i, we should find a way to compensate the mismatch between the target policy π^i and the behavior policy π^{beh}. Uchibe and Doya [28] use an *importance sampling* method for this compensation and demonstrate the implementation over several RL algorithms. Adopting their approach, during each update of π^i when following π^{Beh} we use a compensation factor $IS = \frac{Q^i(s_m,a_m)}{Q^{Beh}(s,a)}$ during Q-learning as depicted in Equation 6, where (s_m, a_m) are *mapped* state-action pairs of (s, a) in behavior policy to attribute i, and α as learning rate.

$$Q^i(s_m, a_m) = Q^i(s_m, a_m) + \alpha \left[R(s_m) + \gamma \cdot IS \cdot \max_{a'}(Q^i(s_m, a')) - Q^i(s_m, a_m) \right] \quad (6)$$

$R(.)$ in the above equation is different from the immediate reward $r(.,.)$ introduced in section 7. In an anticipatory system, we are interested in the impact of future predictions on the current state of the system. This means that the reward for a state-action pair would correspond to future predicted states. With this regard, equation 7 calculates $R(s_t)$ with γ as a discount factor. Future predicted states and actions (s_{t_i}, a_{t_i}) are obtained by applying an ϵ-greedy algorithm on the current policy matrix and starting from s_t.

$$R(s_t) = \sum r(s_t, a_t) + \gamma r(s_{t+1}, a_{t+1}) + \cdots + \gamma^m r(s_{t+m}, a_{t+m}) + \cdots \quad (7)$$

This scheme defines the *collaborative* aspect of interactive learning. For example, during a learning episode, pitch attribute can become the behavior policy Q^{beh} and during that whole episode the system follows the pitch policy for simulations and other attributes' policies $Q^i(.,.)$ will be influenced by the behavior of the pitch policy as shown in equation 6.

8.2 Memory-based Learning

In the Q-learning algorithm above, state-action pairs are updated during each episode through an ϵ-greedy algorithm on previously learned policies and using updated rewards. This procedure updates one state-action pair at a time. In an ideal music learning system, each immediate change should evoke previous related states already stored in the memory. In general, we want to go back in the memory from any state whose value has changed. When performing update, the value of some states may have changed a lot while others rest intact, suggesting that the predecessor pairs of those who have changed a lot are more likely to change. So it is natural to prioritize the backups according to measures of their urgency and perform them in order of priority. This is the idea behind *prioritized sweeping* [29] embedded in our learning with the priority measure as in equation 8 for a current state s and next state s', leading to a priority queue of state-action pairs (chosen by a threshold θ) to be traced backwards for more updates.

$$p \leftarrow |R(s) + \gamma \max_{a'}(Q^{Beh}(s', a')) - Q^{Beh}(s, a)| \quad (8)$$

9 Generation Results

There are many ways to generate or improvise once the policies for each attribute are available. We represent one simple solution using the proposed architecture. At this stage, the system would be in the *self listening mode* (Figure 3(b)). The agent would generate *phrases* of fixed length following a behavior policy (learned from the previous interaction). When following the behavior attribute, the system needs to *map* the behavior state-action pairs to other agents in order to produce a complete music event. For this, we first check and see whether there are any common transitions between original attributes and, if not, we would follow the policy for their derivative behavior. Once a phrase is generated, its (negative) reinforcement signal is calculated and policies are updated as in section 8 but without updating the current models (FOs).

Audio results of automatic improvisation sessions on different styles can be heard at the following URL:

`http://www.crca.ucsd.edu/arshia/ABiALS06/`

As a sample result for this paper, we include analysis of results for style imitation on a polyphonic piece, *two-part Invention* No.3 by J.S. Bach. For this example, the learning phase was run in *interaction mode* with a sliding window of 50 events with no overlaps over the original MIDI score. After the learning phase, the system entered *self listening* mode where it generates sequences of 20 events and reinforces itself until termination. Parameters used for this session were $\alpha = 0.1$ (in eq. 6), $\gamma = 0.8$ (in eq. 7), $\theta = 2$ for prioritized sweeping threshold, and $\epsilon = 0.1$ for the *epsilon*-greedy selection of state-action pairs. Number of episodes simulated during each RL phase was 100. The generated score is shown in Figure 5 for 240 sequential events where the original score has 348. For this generation, the *pitch* behavior has *won* all generation episodes and direct mappings of *duration* and *harmonic* agents have been achieved 76% and 83% in total respectively leaving the rest for their derivative agents.

While both voices follow a polyphonic structure, there are some formal musicological structures that can be observed in the generated score. Globally, there are *phrase* boundaries in measures 4 and 11 which clearly segment the score into three formal sections. Measures 1 to 4 demonstrate some sort of exposition of musical materials which are expanded in measures 7 to the end with a transition phase in measure 5 and 6 ending at a week cadence on `G` (a fourth in the given key). There are several thematic elements which are reused and expanded. For example, the repeated `D` notes appearing in measures 2 appear several times in the score notably in measure 7 as low A with a shift in register and harmony and measure 9 and 15. More importantly, these elements or their variants can be found in the original score of Bach.

Figure 6 shows the pitch-harmony space of both the original MIDI and the generated score. As is seen, due to the collaborative and competitive multi-agent architecture of the system, there are new combinations of attributes which do not exist in the trained score.

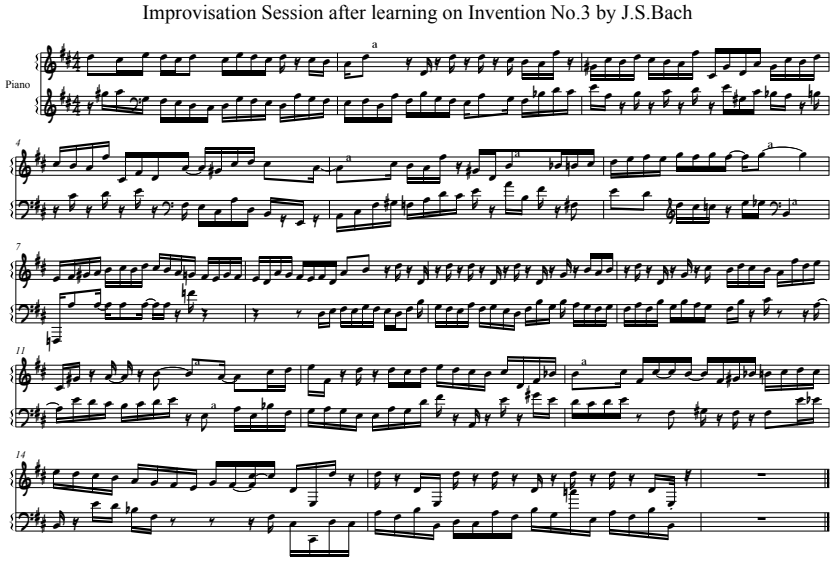


Fig. 5. Style imitation sample result

10 Discussions

In this chapter we presented an anticipatory model of music cognition with application to automatic improvisation and style imitation. The proposed model covers short-term and working memory processes introduced in music cognition literature which result in dynamic adaptive expectations and long-term planning. The anticipatory model, in ABiALS terms, is a payoff and state anticipatory system which provides attentive and preventive frameworks during computation. We show that generation results demonstrate long-term and complex behavior thanks to this anticipatory and cognitive model.

Before any discussion, we would like to bring forth the difficulty of *evaluation* in case of automatic music generation. As should be clear to any musical reader, assessing a music generator in an objective manner, if not impossible, would set along disputable measures of goodness. On the other hand, in most music practices and styles, what is considered as *wrong* can be constituted as a *feature* depending on the context. Therefore we do not discuss the outcome of our design in aesthetical terms. Such considerations might become possible by careful design of perceptual experiments with human subjects which we will address in our future work. Here we discuss further issues such as complexity of the proposed model and further research.

10.1 Model Complexity

In the architecture introduced above, because of the concurrent and competitive multiple agent structure, each component or attribute is modeled separately and

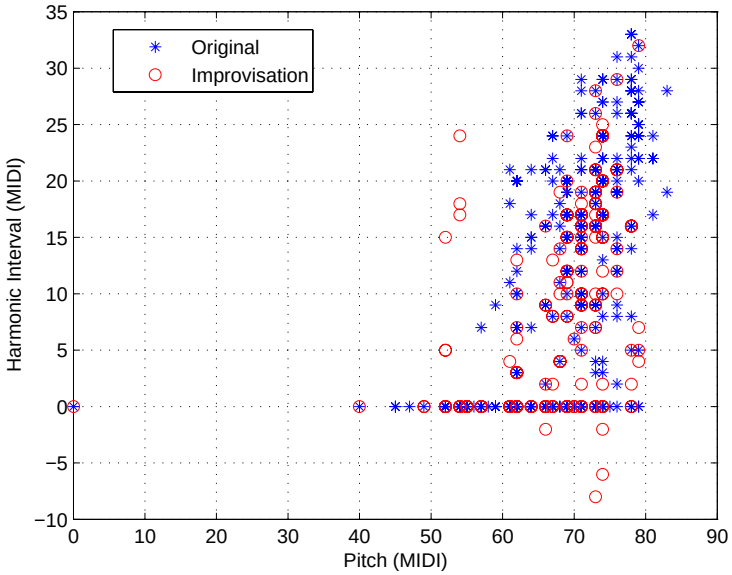


Fig. 6. Improvisation Space vs. Original Space

the state-space size increases linearly with *time* as $k \times T$ coming down to 45 for the toy example. Modal interaction is not modeled by directed graphs between components but rather by influence of each attribute on others through the IS term in equation 6 as a result of collaboration and competition between agents. Note that this choice comes from cognitive foundations of music and was not made for mere simplicity. The complexity of the system depends linearly on T, n_is and an adaptive environmental factor. This is because the arrows of the state-space model are inferred on-line and is dependent on the context being added to previous stored knowledge. We could say that it has an upper-bound of $O(nkT)$ but is usually much sparser than that.

The fact that T is a factor of both state-space size and complexity has advantages and shortcomings. The main advantage of this structure is that it easily enables us to access long-term memory and calculate long-term dependencies, induce structures, and go back and forth in the memory at ease. However, one might say that as T grows, models such as Factorial Markov would win over the proposed model in terms of complexity since n_i would not change too much after some time T. This shortcoming is partially compensated by considering the phenomena of *finite memory process*. A finite memory process in our application is one that, given a factor oracle with N states and an external sequence A^t, can generate the sequence through a finite window of its history without using all its states [30]. More formally, this means that there exist a nonnegative number m such that the set $\{s_n \in FO : n \in \mathbb{N} \text{ and } n \in [N-m, N]\}$ would suffice for

regeneration of a new sequence A^t. This notion is also supported by the fact that music data in general is highly repetitive [3] and not considering this would cause high reinforcement of earlier states in the memory through time. The parameter m is usually dependent on the style of music but for this presentation we keep it fixed.

Besides observing results, compared to similar systems, an anticipatory model reduces the complexity of the representation and learning. The proposed model and shown result need much less training data for learning (a single piece of music as training data to generate a rather long polyphonic sample) and is currently being developed for realtime improvisation.

10.2 Further Developments

As mentioned earlier, an ideal anticipatory model of music should consider all expectation processes in music perception mentioned in section 2. In our first experiment, we attempted two and left the more difficult semantic and episodic processes for later works. To compliment the system, we would need more intelligent modules for music semantic learning and better representational schemes. Note that the sample result in figure 5 is a result of automatic interactive reinforcement learning without explicit consideration for semantic notions such as harmonic progressions or counterpoint. Adding these two notions to the system should further improve local consistencies in the results.

The interactive learning module can still be more efficient in each episode by considering directly relevant states for updates. This will bring us to the notion of Active Learning for future work. Also, note that the representation module using Factor Oracles does not in any ways represent the complexity of feature extraction and perceptual bindings of the auditory system in the brain. It was rather chosen as a very efficient way to gather repetitive factors and structures in a sequence. Further alternatives should be studied for enhancement of this model.

References

1. Leonard B. Meyer. *Emotion and Meaning in Music*. Univ. of Chicago Press, 1956.
2. Martin Butz, Olivier Sigaud, and Pierre Gérard. Internal models and anticipations in adaptive learning systems. In *Anticipatory Behavior in Adaptive Learning Systems*, pages 86–109, 2003.
3. David Huron. *Sweet Anticipation: Music and the Psychology of Expectation*. MIT Press, 2006.
4. J. R. Saffran, E. K. Johnson, R. N. Aslin, and E. L. Newport. Statistical learning of tonal sequences by human infants and adults. cognition. In *Cognition*, volume 70, pages 27–52, 1999.
5. G. Edelman. *Neural Darwinism: The Theory of Neuronal Group Selection*. Basic Books, 1987.
6. Bob Snyder. *Music and Memory: An Introduction*. MIT Press, New York, 2000.
7. Lejaren A. Hiller and L. M. Isaacson. *Experimental Music: Composition with an Electronic Computer*. McGraw-Hill Book Company, New York, 1959.

8. I. Xenakis. *Formalized Music*. University of Indiana Press, 1971.
9. Shlomo Dubnov, G. Assayag, O. Lartillot, and G. Bejerano. Using machine-learning methods for musical style modeling. *IEEE Computer Society*, 36(10):73–80, 2003.
10. Gérard Assayag and Shlomo Dubnov. Using factor oracles for machine improvisation. *Soft Computing*, 8-9:604–610, 2004.
11. François Pachet. The continuator: Musical interaction with style. In *Proc. of International Computer Music Conference*, Gotheborg, Sweden, September 2002.
12. Darrell Conklin and I. Witten. Multiple viewpoint systems for music prediction. In *Journal of New Music Research*, volume 24, pages 51–73, 1995.
13. Lawrence K. Saul and Michael I. Jordan. Mixed memory markov models: Decomposing complex stochastic processes as mixtures of simpler ones. *Machine Learning*, 37(1):75–87, 1999.
14. Darrell Conklin. Music generation from statistical models. In *Proceedings of Symposium on AI and Creativity in the Arts and Sciences*, pages 30–35, 2003.
15. Jacob Ziv and Abraham Lempel. Compression of individual sequences via variable-rate coding. *IEEE Transactions on Information Theory*, 24(5):530–536, 1978.
16. M. Feder, N. Merhav, and M. Gutman. Universal prediction of individual sequences. *IEEE Trans. Inform. Theory*, 38(4):1258–1270, Jul 1992.
17. Shlomo Dubnov, R. El-Yaniv, and Gérard Assayag. Universal classification applied to musical sequences. In *Proc. of ICMC*, pages 322–340, Michigan, 1998.
18. Dana Ron, Yoram Singer, and Naftali Tishby. The power of amnesia: Learning probabilistic automata with variable memory length. *Machine Learning*, 25(2-3):117–149, 1996.
19. Cyril Allauzen, Maxime Crochemore, and Mathieu Raffinot. Factor oracle: A new structure for pattern matching. In *Proc. of Conference on Current Trends in Theory and Practice of Informatics*, pages 295–310, London, 1999. Springer-Verlag.
20. John A. Biles. Genjam in perspective: A tentative taxonomy for genetic algorithm music and art systems. *Leonardo*, 36(1):43–45, 2003.
21. Judy A. Franklin. Predicting reinforcement of pitch sequences via lstm and td. In *Proc. of International Computer Music Conference*, Miami, Florida., 2004.
22. Marcus Pearce, Darrell Conklin, and Geraint Wiggins. Methods for combining statistical models of music. In U. K. Wiil, editor, *Computer Music Modelling and Retrieval*, pages 295–312, 2004.
23. Robert Rosen. *Anticipatory Systems*, volume 1 of *IFSR International Series on Systems Science and Engineering*. Pergamon Press, Oxford, 1985.
24. Paul Davidsson. A framework for preventive state anticipation. In *Anticipatory Behavior in Adaptive Learning Systems*, pages 151–166, 2003.
25. Shlomo Dubnov. Spectral anticipations. *Computer Music Journal*, 2006.
26. Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.
27. A. Lefebvre and T. Lecroq. Computing repeated factors with a factor oracle. In *Proc. of the Australasian Workshop On Combinatorial Algorithms*.
28. E. Uchibe and K. Doya. Competitive-cooperative-concurrent reinforcement learning with importance sampling. In *Proc. of International Conference on Simulation of Adaptive Behavior: From Animals and Animats*, pages 287–296, 2004.
29. Andrew Moore and Chris Atkeson. Prioritized sweeping: Reinforcement learning with less data and less real time. *Machine Learning*, 13:103–130, 1993.
30. Alvaro Martin, Gadiel Seroussi, and J. Weinberger. Linear time universal coding and time reversal of tree sources via fsm closure. *Information Theory, IEEE Transactions on*, 50(7):1442–1468, july 2004.