A Coupled Duration-Focused Architecture for Real-Time Music-to-Score Alignment

Arshia Cont, *Student Member*, *IEEE*

Abstract—The capacity for real-time synchronization and coordination is a common ability among trained musicians performing a music score that presents an interesting challenge for machine intelligence. Compared to speech recognition, which has influenced many music information retrieval systems, music's temporal dynamics and complexity pose challenging problems to common approximations regarding time modeling of data streams. In this paper, we propose a design for a real-time music-to-score alignment system. Given a live recording of a musician playing a music score, the system is capable of following the musician in real time within the score and decoding the tempo (or pace) of its performance. The proposed design features two coupled audio and tempo agents within a unique probabilistic inference framework that adaptively updates its parameters based on the real-time context. Online decoding is achieved through the collaboration of the coupled agents in a Hidden Hybrid Markov/semi-Markov framework, where prediction feedback of one agent affects the behavior of the other. We perform evaluations for both real-time alignment and the proposed temporal model. An implementation of the presented system has been widely used in real concert situations worldwide and the readers are encouraged to access the actual system and experiment the results.

Index Terms—Automatic musical accompaniment, hidden hybrid Markov/semi-Markov models, computer music.

1 Introduction

Real-time alignment of audio signals to symbolic music scores, or score following, has a long tradition of research dating back to 1983 [1], [2]. Both the original and current motivations behind score following consist of live synchronization between a computer with a symbolic music score and a musician performing the same score with a musical instrument. This can also be extended to a live computer accompaniment with a human performer, where the computer assumes the performance of the orchestral accompaniment while the human performs the solo part. Another musical motivation is new music repertoire, primarily live electronic performances in which the computer performs a live electronic score that should be synchronous to the human performer in a real-time performance situation.[1] Our overall intention is to bring the computer into the performance as an intelligent and well-trained musician capable of imitating the same reactions and strategies during music performance that human performer(s) would undertake. In recent years, automatic audio to score alignment systems has become popular for a variety of other applications, such as Query-by-Humming [3], intelligent audio editors [4], and as a front end for many music information retrieval systems.

A minimal description of real-time score following is as follows: The system possesses a representation of the symbolic music score in advance, which is given by the user and fed into the system offline. The goal of the system is to map the incoming real-time audio stream onto this representation and decode the current *score position* and *real-time tempo* (the dynamic musical clock used by musicians, to be defined shortly). Fig. 1 demonstrates this through an excerpt of a music score on the top, the real-time setup scheme of the live performance in the middle, and sample results from the alignment of a recording audio onto the excerpt score indicating decoded event positions and timing information.

This paper proposes an architecture for the modeling of the temporal dynamics of music events on the fly through parallel decoding of two coupled audio and tempo agents, which could be extended to similar applications in other domains. In most transcribed musical cultures (including western musical notation), time is usually written with values relative to a musical clock referred to as the *tempo*. Tempo in western cultures is usually indicated by the number of beats that is expected to occur in a minute (BPM) and accordingly, the temporality of events in a symbolic score is indicated by the number of expected beats that they should span in time, which can be fractions of a pulse. The dynamic variation of tempo, or the musical clock, is highly responsible for musical expressivity among musicians and could lead to extreme variations in the distribution density of an underlying generative model of audio. The alignment problem presented in this paper is similar to well-studied problems from speech recognition and segmentation such as speech-to-phoneme or speech-to-text alignment [5]. Most real-time systems for speech applications use generative models of the audio signal using hidden Markov models (HMMs). One of the main issues with this type of generative model for speech and audio has been to accurately model duration distributions of the underlying events, leading to variants of HMM. In most approaches, the generative models along their underlying parameters describing (implicit or explicit) duration models are obtained through offline learning, thus assuming stationarity of the input data

1. See http://imtr.ircam.fr/index.php/Score_Following/ for demonstrative videos.

• The author is with the Ircam-Centre Pompidou, 1 place Igor Stravinsky, Paris 75004, France. E-mail: arshia.cont@ircam.fr.

Manuscript received 17 Aug. 2008; revised 21 Feb. 2009; accepted 22 Apr. 2009; published online 5 May 2009.
Recommended for acceptance by V. Pavlovic.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2008-08-0543.
Digital Object Identifier no. 10.1109/TPAMI.2009.106.

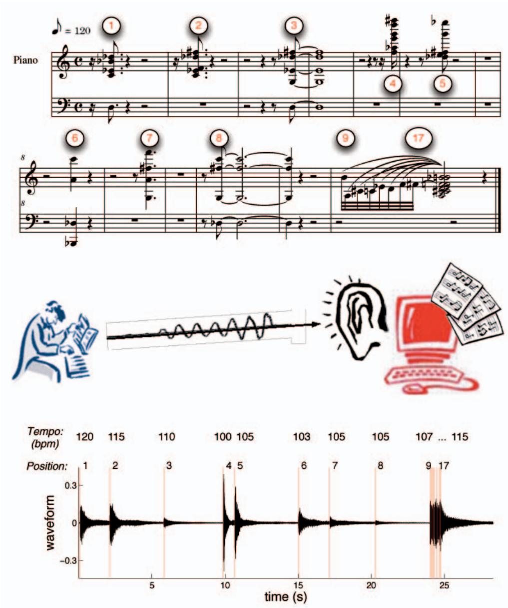

Fig. 1. General schema of a score-following application.

with regard to the learned models. Recent experiments in [6] show that by using standard HMMs with an increased number of states for each symbol (e.g., phonemes), we are capable of closely matching the performance of duration-focused approaches such as semi-Markov models or variable transition probabilities as applied to classical speech problems. Given the extreme variability of temporal dynamics of musical events, such approximations would lead to shortcomings in the performance of the system; therefore, we assume that the temporal structure of underlying events is a dynamic structure. The models proposed in this paper are capable of decoding such temporal dynamics on the fly and could be extended to other domains.

Our proposed method is based on an anticipatory forward propagation algorithm for real-time inference of event alignment and tempo parameters. The problem is similar in nature to finding the most optimal path within a sequence (e.g., music events in our case). This is usually addressed using the Viterbi algorithm [7], which uses both forward and backward propagation of beliefs at each time node t to decode the optimal path. In a real-time and reactive context such as ours, the system has no access to backward beliefs as the outcomes of future observations. In such situations, researchers usually rely on forward propagation to decode the optimal position or by cascading other sources of information (through joint or independent distributions). In our architecture, the absence of future observations is compensated by formulating the problem as in anticipatory systems. An *Anticipatory System* is "a system containing a predictive model of its environment, which allows it to change state at an instant in accordance with the model's predictions pertaining to a later instant" [8]. In short, anticipatory behavior can be artificially obtained by a feedback of the system's prediction into the future in order to affect current time decisions. In this paper, we focus on *state anticipation*, where explicit predictions of state durations for time $T > t$ affect the state of decision at time $T = t$.

In this paper, we present a real-time and online audio to score alignment system that is also capable of decoding the live *tempo* (or musical pace) of the performance. In its design and in comparison to existing systems, the proposed system encompasses two coupled audio and tempo agents that collaborate and compete in order to achieve synchrony. Collaboration is achieved through the feedback of prediction of one agent into the other. The inference engine features a hidden hybrid Markov/semi-Markov model that explicitly models events and their temporalities in the music score. The tempo agent features a self-sustained oscillator based on [9], adopted here in a stochastic framework for audio processing. The novelty of the proposed design is twofold: 1) coupling of two parallel audio and tempo agents through a unique real-time inference technique and 2) online adaptation of system parameters (duration distributions) to the real-time context, leaving no need for offline training and leading to global reduction of learned parameters compared to existing systems. The system gets as input a score representation and an audio stream. The outputs of the system are the event indexes and real-time tempo, with no need for external training. In practice, the presented system is capable of successfully decoding polyphonic music signals and has been featured in several concert performances worldwide with various artists including a performance with the Los Angeles Philharmonic.[2]

The paper is organized as follows: We introduce the research background on the topic in Section 2.1 as well as background information on the foundations of musical time in Section 2.2 that has inspired our computational approach. We introduce the general architecture as well as the employed sequential modeling in Section 3 and provide the general inference framework thereafter in Section 4. Sections 5, 6, and 7 detail the generative models and modeling aspects of the inference framework. We evaluate the system's performance in various situations in Section 8, followed by discussion and conclusion.

2 Background

2.1 Score Following Research

Score following research was first introduced independently in [1] and [2]. Due to the computational limitations at the time, both systems relied on symbolic input through sensors installed on the musical instrument rather than raw audio. The problem would then be reduced to *string matching* between the symbolic input stream and the score sequence in real time. The issue becomes more complicated when expressive variations of the performer, either temporal or event-specific, come into play. Underlying systems must have the tolerance to deal with these variations as well as human or machine observation errors in order to remain synchronous with the human performer. All of these factors made the problem challenging even on the symbolic level.

In the early 1990s, with the advent of faster computers, direct use of audio input instead of symbolic data became possible, allowing musicians to use their original instruments. In this new framework, the symbolic level of the score is not directly observable anymore and is practically *hidden* from the system. Early attempts used monophonic pitch detectors on the front end, providing pitch information to the matching algorithm under consideration, thereby doubling

2. Performance of *Explosante-Fixe* by composer Pierre Boulez, LA Philharmonic, Disney Hall, Los Angeles, 13 and 14 Jan. 2008.

the problem of tolerance with the introduction of pitch detection uncertainly which is an interesting problem by itself (e.g., [10]). By the mid-1990s, in parallel with developments made in the speech recognition community, Grubb and Dannenberg [11] and Raphael [12] introduced the stochastic approach. The latter is based on Hidden Markov Models and statistical observations from live audio input. Raphael's approach was further developed by various authors, leading to variants (for example, [13], [14]). In a more recent development, Raphael introduced a polyphonic alignment system where tempo is decoded along score positions [15]. This design has two (cascaded) stages for decoding score position and tempo. The first stage consists of a Hidden Markov Model deduced from the score, which is responsible for decoding the position in the score (called the *listener*). The second stage uses an elaborate Bayesian network to deduce the smooth tempo during the performance. In this paper, we propose an anticipatory model for the problem of score following in which tempo and audio decoding are not separate problems but are coupled together within a single framework.

Offline versions of score following, where the whole audio sequence is entirely known prior to actual synchronization, have been vastly studied in the literature (see [16, Chapter 5] and references therein). Many of these systems make use of generative models such as HMMs or their variants such as Dynamic Time Warping algorithms. In this paper, we focus on the online and real-time problem, where the audio streams arrive incrementally into the system.

2.2 Foundation of Musical Time

The perception of musical metric structure in time is not merely an analysis of rhythmic content; rather, it shapes an *active listening* strategy in which the listener's expectations about future events can play a role as important as the musical events themselves. The assumptions inherent in this imply that, contrary to basic speech to phoneme applications, the temporal structure of musical expectation is a dynamic structure and should be handled at the onset of the design. This fact is further enhanced by observing the manner in which various cultures have managed to transcribe dynamic temporal structures of music through music notation. Looking at a traditional western notated music score, the simplest way to transcribe temporal dynamics would be a set of discrete sequential notes and silence events that occupy a certain amount of relative duration in time. The relative notion of musical time is one of the main sources of musical expressivity, which is usually guided by *tempo*, often represented in beats per minute (BPM). While limiting our discourse to the realm of western classical music notation, we provide two important insights on the temporality of musical events with direct consequences on our model, as expressed by two important figures of late 20th century music composition in [17], [18].

2.2.1 Temporal versus Atemporal

An atemporal (or out-of-time) event corresponds to an object that possesses its own internal temporal structure independent of the overall temporal structures of the piece of music. The two structures are usually considered independent in music theory. To conform this distinction with our probabilistic design, we define an atemporal object or event as one that possesses a physical space in the score but does not contribute to the physical musical time of the score. Typical examples of atemporal objects are grace notes, internal notes of a trill, or typical ornaments in a baroque-style interpretation in western classical notation. In such cases, the individual events do not contribute to the notion of the tempo, but their relative temporal appearance in the case of the grace note, or their overall in-time structure in the case of a trill, contributes to the notion of the tempo.

2.2.2 Striated Time versus Smooth Time

Striated time is one that is based on recurring temporal regularities, while smooth time is a continuous notion of time as a flow of information. The pulsed-time used in most western classical music notation is a regulated striated timeflow which uses an internal musical clock usually driven by a tempo parameter in beats per minute. In our terminology, we distinguish between a striated timescale, where the notion of time is driven relative to a constantly evolving tempo, and a smooth timescale, where the information on the microscopic level consists of individual atemporal elements or is defined relative to a pulse. A typical example of a smooth-time event in western traditional notation is the free *glissandi*. It is important to mention that most *available* classical and popular music pertains to striated time.

2.3 Probabilistic Models of Time

Because of the intrinsic temporal nature of music, the ability to represent and decode temporal events constitutes the core of any score following system. In general, a live synchronization system evaluates live audio inputs versus timed models of a symbolic score in its memory. Since such systems are guaranteed to operate in uncertain situations, probabilistic models have become a trend in modeling since the late 1990s. Within this framework, the goal of a probabilistic model is to decode the temporal dynamics of an outside process. Therefore, the performance of such models is highly dependent on their ability to represent such dynamics within their internal topology. In these problems, any state of a given process occupies some duration that can be deterministic or not. We are interested in a probabilistic model of the macrostate duration and expected occupancy. In a musical context, a macrostate can refer to a musical event (note, chord, silence, trills, etc.) given an expected duration that might be composed of one or more microstates. A common way to model time-series data in the literature is by the use of *state-space models*. A state-space model of a sequence is a time-indexed sequence of graphs (nodes and edges), where each node refers to a state of the system over time. Therefore, each state has an explicit time occupancy that can be used to probabilistically model the occupancy and duration of the events under consideration. In this section, we limit our study to two wide classes of state-space models and their duration models, which cover most existing approaches: Markov and semi-Markov processes.

2.3.1 Markov Time Occupancy

In a parametric Markov time model, the expected duration of a macrostate j (events such as notes, chords, etc., that occupy time) is modeled through a set of Markov chains (or microstates) with random variables attached to transition probabilities, which parameterize an occupancy distribution $d_j(u)$, where the random variable U accounts for the

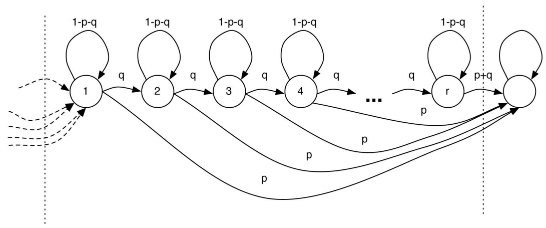

Fig. 2. Parametric Markov topology.

number of times spent in the macrostate j. Fig. 2 shows a parametric macrostate Markov chain topology commonly used for duration modeling. This way, the macrostate consists of r Markov states and two free parameters p and q corresponding, respectively, to the exit probability and the next-state transition probability. The macrostate occupancy distribution associated with this general topology is the compound distribution:

$$P(U=u)=\sum_{n=1}^{r-1}\binom{u-1}{n-1}(1-p-q)^{u-n}q^{n-1}p + \binom{u-1}{r-1}(1-p-q)^{u-r}q^{r-1}(p+q).$$

If $p=0$, this macrostate occupancy distribution is the negative binomial distribution:

$$P(U=u)=\binom{u-1}{r-1}q^{r}(1-q)^{u-r},$$

which corresponds to a series of r states with no jumps to the exit state with the shortcoming that the minimum time spent in the macrostate is r. This simplified version has been widely explored in various score following systems, where the two parameters r and q are derived by optimization over the macrostate's time duration provided by the music score [12], [13].

2.3.2 Semi-Markov Time Occupancy

In a Semi-Markov model, a macrostate can be modeled by a *single* state (instead of a fixed number of microstates) and by using an explicit time occupancy probability distribution $d_j(u)$ for each state j and occupancy u. Assuming that S_i is the discrete random variable denoting the macrostates at time i from a state space $\mathcal{S}\subset\mathbb{N}$ and T_m is the time spent at each state m, then $S_t=m$ whenever

$$\sum_{k=1}^{m}T_k\le t<\sum_{k=1}^{m+1}T_k.$$

Or simply, we are at state m at time t when the duration models for all states up to m and $m+1$ comply with this timing. In this configuration, the overall process is *not* a Markov process within macrostates but rather a Markov process in between macrostates, hence the name *semi-Markov*.

The explicit occupancy distribution can then be defined as follows:

$$d_j(u)=P(S_{t+u+1}\ne j, S_{t+u-v}=j, v\in[0,u-2]\,|\,S_{t+1}=j, S_t\ne j), \quad (1)$$

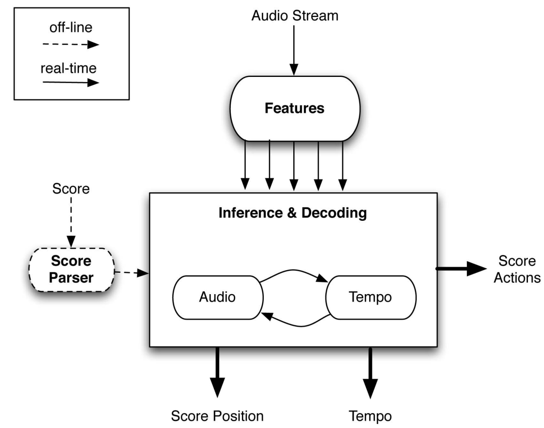

Fig. 3. General system diagram.

where $u=1,\ldots,M_j$, with M_j the upper bound for the time spent in the macrostate.

Semi-Markov models were first introduced in [19] for speech recognition and gained attention because of their intuitive access to models' temporal structure. Semi-Markov topologies are usually much more sparse in computations and controllable than their Markovian counterparts. Moreover, they provide explicit access to time models expressed as occupancy distributions. Despite these advantages, explicit duration models might require substantial development of standard statistical inference algorithms. Such developments could become cumbersome if duration models are assumed to be dynamic (as is the case in our framework) rather than stationary (as with most speech recognition problems).

3 General Architecture

The general description of our score following task is as follows: Having possession of the symbolic music score in advance, the goal of our system is to map the incoming real-time audio stream onto this representation and decode the current *score position*, *real-time tempo,* and undertake *score actions*. In this paper, we focus on the first two (as demonstrated in Fig. 1). *Score actions* involve musical programming either for live electronics effects or automatic accompaniment applications and are reported in [20]. The music score is represented by a probabilistic state-space model constructed directly from a symbolic music score inspired by the observations in Section 2.2. Given the score's state-space representation, the real-time system extracts instantaneous beliefs or observation probabilities of the audio features calculated from the stream, with regard to the states of the score graph. The goal of the system is to then integrate this instantaneous belief with past and future beliefs in order to decode the position and tempo in real time. Fig. 3 shows a general diagram of our system.

To tackle the problem, we adopt a generative approach with the underlying hypothesis that the audio signal can be generated by the underlying state-space score model. Formally speaking, we assume that the audio features through time τ or x_0^τ (short for $x_0,\ldots,x_\tau$) are stochastic processes represented by the random variable $\{X_t\}$, which is generated by a sequence of states s_0^τ through the random variable $\{S_t\}$ that describes the symbolic score sequence.

Hence, the problem of score following is the inverse of this hypothesis: to find the most likely state sequence associated with the observed real-time audio sequence. Due to the nature of this inverse problem, the underlying state sequence that generates the audio is not directly observable by the system and is thus *hidden*. This process of finding the most likely state sequence in a hidden process up to the present is referred to as the *inference* problem.

The proposed inference framework, detailed in Section 4, is based on two parallel and coupled audio and tempo agents, and adaptively handles the temporal dynamics of musical structures. The two agents collaborate at all times to map the real-time audio input to the most likely state sequence in the score model. This choice of design is motivated by strong evidence in brain organization for music processing for dissociation of pitch and temporal processing of music [21]. This evidence suggests that these two dimensions involve the operation of separable neural subsystems and thus can be treated in parallel in a computational framework. The tempo agent computes on the event timescale as provided by the audio agent and is based on a cognitive model of musical metric structure and provides continuous tempo predictions based on live audio input, detailed in Section 7.2. Besides decoding real-time tempo, the tempo agent dynamically assigns the duration distributions used for score position alignment, as detailed in Section 7.3. The inference scheme computes on the *continuous* audio timescale and assigns probabilistic values to relevant states in the score state-space by combining tempo predictions and continuous audio observations. The proposed model is thus an anticipatory and coupled system, where the state likelihoods are influenced dynamically by the predicted tempo, and in return, the tempo agent is directly affected by the instantaneous alignment positions through audio decoding.

The state-space generative model of the score proposed here is a *Hidden Hybrid Markov/semi-Markov chain* [22], which is motivated by observations in Section 2.2 and probabilistic models of time, as defined in the following section.

3.1 Hybrid Models of Time

The probabilistic (and generative) state-space model of the score describes the event types and time models of events in the score, which are used during decoding and inference. For the state-space model of our framework, we propose using the best of both of the probabilistic time models presented previously in Section 2.3, motivated by the observations on compositional foundations of time as described in Section 2.2. Within this framework, we would like to take advantage of explicit time models of semi-Markov chains for *temporal* and *striated-time* events, and employ parametric Markov models for *atemporal* and *smooth-time* elements in a music score. These considerations lead to a probabilistic model based on *Hybrid Markov/semi-Markov Chains* as proposed in [22]. In this section, we provide a formal definition of this model and detail its construction from a music score in Section 5.

Following our previous formalization, we then assume that the audio features represented by the random variable $\{X_t\}$ are generated by a sequence of states through the state process $\{S_t\}$ corresponding to (the hidden) states in a hybrid Markov/semi-Markov chain constructed from the score. A discrete hidden hybrid Markov/semi-Markov chain can then be viewed as a pair of stochastic processes (S_t, X_t), where the discrete output $\{X_t\}$ is related to the state process $\{S_t\}$ by a stochastic function denoted by f, where $X_t = f(S_t)$. Since this mapping f is such that $f(s_j) = f(s_k)$ may be satisfied for different j and k, or in other words, a given output may be observed in different states, the state process S_t is not observable directly but only indirectly through the output process X_t. Beyond this point, we use $P(S_t = j)$ as shorthand for $P(S_t = s_j)$, which denotes the probability that state j is emitted at time t.

Let S_t be a J-state hybrid Markov/semi-Markov chain. It can then be defined by the following:

- Initial probabilities $\pi_j = P(S_0 = j)$ with $\sum_j \pi_j = 1$.
- Transition Probabilities:
  - semi-Markovian state j and $\forall j, k \in \mathbb{N},\ k \neq j$:
  $$p_{jk} = P(S_{t+1} = k | S_{t+1} \neq j, S_t = j),$$
  where $\sum_{k \neq j} p_{jk} = 1$ and $p_{jj} = 0$.
  - Markovian state j:
  $$\tilde{p_{jk}} = P(S_{t+1} = k | S_t = j)$$
  with $\sum_k \tilde{p_{jk}} = 1$.
- An *explicit* occupancy distribution attached to each semi-Markovian state as in (1). Hence, we assume that the state occupancy distributions are concentrated on finite sets of time points.
- An *implicit* occupancy distribution attached to each Markovian state j, where
$$P(S_{t+1} = k | S_{t+1} \neq j, S_t = j) = \frac{\tilde{p_{jk}}}{1 - \tilde{p_{jk}}}$$
defines an implicit state occupancy distribution as the geometric distribution with parameter $1 - \tilde{p_{jk}}$:
$$d_j(u) = (1 - \tilde{p_{jk}})\tilde{p_{jk}}^{u-1}. \qquad (2)$$

The output (audio) process X_t is related to the hybrid Markov/semi-Markov chain S_t by the observation or emission probabilities

$$b_j(y) = P(X_t = y | S_t = j) \quad \text{where} \quad \sum_y b_j(y) = 1.$$

This definition of the observation probabilities expresses the assumption that the output process at time t depends only on the underlying hybrid Markov/semi-Markov chain at time t.

The original formulations of the hybrid network defined above in [22] are not aimed at real-time decoding, neither anticipatory nor multiagent processing. In the following sections, we extend this framework to our coupled anticipatory framework.

4 Inference Formulation

The solution to the inference problem determines the most likely state sequence S_0^τ that would generate X_0^τ, and in the process, the score position and real-time decoded tempi. In a non-real-time context, an exact inference can be obtained using a Viterbi-type algorithm [23] that for each time t uses both beliefs from time 0 through τ (referred to as *forward* *propagation* or $\alpha(t)$) and future knowledge from present (τ) to a terminal state at time T (referred to as *backward propagation* or $\beta(t)$). In a score following system that necessitates on-the-fly synchronization of audio with the music score since using the backward propagation of the Viterbi algorithm is either impossible or would introduce considerable delays in the system. In the proposed system, we hope to compensate for this absence of future beliefs with our anticipatory model of audio/tempo coupled agents and an adaptive *forward* propagation procedure. Here, we formulate a dynamic programming approach for an adaptive forward propagation for a hidden hybrid Markov/semi-Markov process.

For a semi-Markovian state j, the Viterbi recursion of the forward variable is provided by the following dynamic programming formulation (see the Appendix for derivation):

$$\begin{aligned}\alpha_j(t) &= \max_{s_0,\ldots,s_{t-1}} P\big(S_{t+1} \neq j, S_t = j, S_0^{t-1} = s_0^{t-1}, X_0^t = x_0^t\big)\\ &= b_j(x_t)\\ &\times \max\left[\max_{1\leq u\leq t}\left(\left\{\prod_{v=1}^{u-1} b_j(x_{t-v})\right\} d_j(u) \max_{i\neq j}(p_{ij}\alpha_i(t-u))\right)\right].\end{aligned} \quad (3)$$

For a Markovian state j, the same objective amounts to [7]:

$$\begin{aligned}\tilde{\alpha}_j(t) &= \max_{s_0,\ldots,s_{t-1}} P\big(S_t = j, S_0^{t-1} = s_0^{t-1}, X_0^t = x_0^t\big)\\ &= b_j(x_t) \max_i \big(\tilde{p_{ij}}\tilde{\alpha}_i(t-1)\big).\end{aligned} \quad (4)$$

Within this formulation, the probability of the observed sequence $x_0^{\tau-1}$ along with the most probable state sequence is $\mathrm{argmax}_j\, [\alpha_j(\tau-1)]$.

In order to compute (4) and (3) in real time, we need the following parameters:

- State types and topologies, which determine the type of decoding and transition probabilities p_{ij}. This probabilistic topology is constructed directly from the music score and is discussed in Section 5.
- Observations probabilities $b_j(x_t)$, which are obtained from real-time audio features (x_t) and are discussed in detail in Section 6.
- The occupancy distribution $d_j(u)$, which decodes and models the musical *tempo* in real time, and the upper bound u of the product in (3), which are discussed in Section 7.
- A prior belief (or belief at time zero), denoted by $\alpha_j(0)$, which is usually assigned to the corresponding starting point on the score during a performance.

The complexity of this propagation procedure is $O(J\tau(J+\tau))$-time in the worst case and $O(J\tau)$-space, where J is the number of states present in the system under consideration and τ the discrete-time element. In a real-time application on a left-right state-space time structure, we can suitably limit J and τ to small homogeneous zones in space and time during filtering as function of the latest decoding position.

5 Music Score Model

Using the inference formulation above, each audio observation is mapped to a state-space representation of the music score, where each event in the score is modeled as one or more states s_j with appropriate characteristics. The state-space in question would be a hidden hybrid Markov/semi-Markov model constructed out of a given music score during parsing. The type of the state (Markov or semi-Markov), its topology, and associated symbols are decided based on the musical construct taken from the music score. In this section, we describe a set of topologies that was designed to address most temporal structures in western music notation as outlined in Section 2.2. In the figures that follow, Markov states are demonstrated by regular circles, whereas semi-Markov states are denoted by double-line circles.

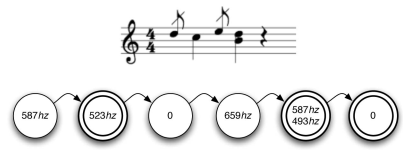

Fig. 4. Sample state-space topology for basic events.

5.1 Basic Events

A single event can be a single pitch, a chord (a set of pitches occurring all at once), or a silence. These events can be either temporal or atemporal (see Section 2.2). A timed event is mapped to semi-Markov state, where an atemporal event (such as a grace note) is mapped to a Markov state. A semi-Markov state s_i is described by a set $\{i, \ell_i, f0_i\}$, where i is the event number or discrete location since the beginning of the score, ℓ_i is its duration expressed as the number beats relative to the initial score tempo, and $f0_i$ is a list of expected pitch frequencies. Fig. 4 shows a sample graphical score and its equivalent Markov topology after parsing. If the duration associated with a single event is set to 0.0, it is a sign that the associated event is atemporal (and therefore, Markovian) and described by $\{i, f0_i\}$. In Fig. 4, grace notes are encoded as Markovian states (circles), where timed pitches are parsed into semi-Markovian (dashed circle) states. In this example, pitches are represented with their fundamental frequencies in hertz and a left-right Markov topology in one-to-one correspondence with the score. Note that in this example, a *dummy* atemporal silence is created in the middle. The parser automatically puts dummy silences between events where appropriate to better model the incoming audio.

5.2 Special Timed Events

Many score models for alignment purposes stop at this point. However, music notation utilizes a large vocabulary in which events are sometimes spread erratically over time and interpretations are either varied from performance to performance or are free at large. This is the case with almost every written music piece that contains events such as *trills* and *glissandos*. While studying some of these common irregularities, we figured out that the particularity of such events is in how they are spreaded over time and how their observations are handled during real-time decoding. We came out with two simple state topologies that address several classical cases as well as more general ones, which are described hereafter. Another motivation for this part of our work is the use of our system by contemporary music composers, who always seek to expand traditional notions of music writing.

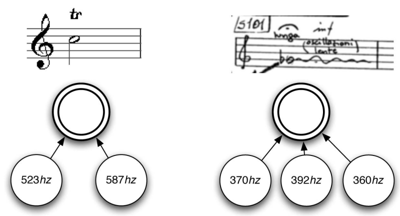

Fig. 5. State-space topology for the TRILL class.

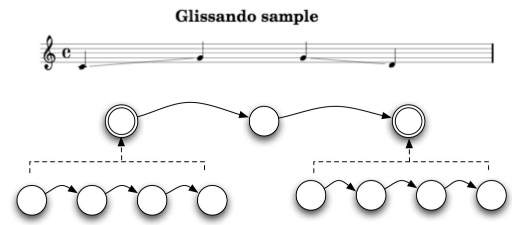

Fig. 6. State-space topology for the MULTI class.

5.2.1 TRILL Class

As the name suggests, the TRILL class is a way to imitate classical music trill notation. In terms of modeling, a trill is one *in-time* event that encompasses several *out-of-time* events. Moreover, time-order, time-span, and the number of repetitions of these substates are of no importance. For example, a whole-tone trill on a middle C with a duration of one beat ($\ell = 1$) can consist of 4 *crotchets*, or 8 *quavers*, or 16 *semiquavers*, etc., of sequences of C and D, depending on the musician, music style, or dynamics of the performance. To compensate for these effects, we consider the TRILL class as one semi-Markov state s_i with a given duration, whose observation $f0_i$ is shared by two or more atemporal states. During real-time decoding, the observation of the general TRILL state is the maximum observation among all possibilities for the incoming audio frame or $b_j(x_t) = \max_{p_i}\{p_i = p(x_t|f0^i_j)\}$. Fig. 5 shows two musical examples that can be described using the TRILL class, where the second[3] demonstrates a *free* glissandos that can also be successfully encoded using this model.

5.2.2 MULTI Class

Another less common situation, but of interest to our applications in music notation, is continuous time events, where the time span of a single event undergoes change in the observation. An example of this in western classical notation is the continuous *glissando* or *portamento*, described as continuously variable pitch, where the musical instrument allows such notations (such as violin, trombone, and the human voice). Moreover, this class of objects would allow matching for continuous data such as audio and gesture, along with symbolic score notations. To this end, we add the MULTI class, which is similar to the TRILL class with the exception that the symbols defined within its context are atemporal Markov states that are *ordered in time*. In this new topology, a high-level semi-Markov state represents the overall temporal structure of the whole object that is mapped to a series of sequential left-right Markov chains. Fig. 6 shows a MULTI example for two consecutive notated glissandi.

6 Observation Model

The inference formulation of Section 4 attempts to map audio signals as discrete frames x_t in time to their corresponding state s_t in a music score. As mentioned before, in our problem, the states are not directly observable by the system, and thus, are hidden. The observation probabilities $b_j(x_t)$ in the inference formulation are thus the *eye* of the system toward the outside world and provide probabilities that the observation vector x_t is emitted from state j. In other words, they are the likelihood probabilities $p(x_t|s_j)$ which, after entering the forward propagation, become posterior beliefs $p(s_j|x_1, x_2, \ldots, x_t)$. The audio stream x_t in our system corresponds to sampled audio signals over overlapping windows of fixed length over time, where t refers to the center of the time window. In the experiments shown in this paper, the time window has a length of 92 ms with an overlap factor of 4 as a compromise between the frequency and time resolution of the input. In this section, we show the model that provides the observation probabilities $b_j(x_t)$ during real-time inference.

In a polyphonic music setting, the observation probabilities should reflect instantaneous pitches that are simultaneously present in an analysis window entering the system in real time. Polyphonic pitch detection is a difficult problem in itself. In our setting, the problem is less complicated since the music score provides prior information regarding expected pitches during the performance. Thus, the goal is to compute the conditional probabilities $p(x_t|s_j)$, where each state s_j provides the expected pitches in the score.

For this aim, we choose to represent analysis frames x_t in the frequency domain using a simple FFT algorithm and compare the frequency distribution to frequency templates constructed directly out of the pitch information of s_j. This choice of observation model is natural since musical pitches tend to preserve quasistationary frequency distributions during their lifetime, which correspond to their fundamental frequencies along with several harmonics. Since we are dealing with x_t and s_t as probability distributions over the frequency domain, it is natural to choose a comparison scheme based on probability density distances for which we choose the Kullback-Leibler divergence as shown below:

$$D(S_j \| X_t) = \sum_i S_j(i) \log \frac{S_j(i)}{X_t(i)}, \tag{5}$$

where X_t is the frequency domain representation of x_t or $\mathcal{FFT}(x_t)$ and S_j is the frequency probability template corresponding to pitches in s_j. Note that the KL divergence of (5) is not a distance metric and is employed as a likelihood observation function: If S_j is considered as the "true" frequency distribution of pitches in s_j and X_t as an approximation candidate for S_j, then $D(S_j \| X_t)$ gives a measure up to which X_t can describe S_j and is between 0 and $+\infty$ with $D(S_j \| X_t) = 0$ iff $S_j = X_t$. To convert (5) to probabilities, we pass it through an exponential function that maps $[0, +\infty] \to [1, 0]$:

$$p(x_t|s_j) = \exp\left[-\beta D(S_j \| X_t)\right], \tag{6}$$

3. The handwritten score excerpts are from the piece *"little i"* for flute and electronics by Marco Stroppa, with kind permission from Casa Ricordi, Milan.

where β is the scaling factor that controls how fast an increase in distance translates to decrease in probability, fixed to 0.5 in our system.

In order to construct the "true" frequency distributions of pitches in s_j, we correctly assume that a pitch consists of a fundamental and several harmonics representing themselves as peaks in the frequency domain. Each peak is modeled as Gaussian, centered on the fundamental and harmonics and their variance relative to their centers on a logarithmic musical scale. We fix the number of harmonics for these templates to 10 for each fundamental with a variance of a halftone in the tempered musical system, which can be adjusted if needed by the user.

Note that the likelihood in (6) requires normalization of X_t and S_j such that they would add to 1. This normalization undermines the robustness of the system to low-energy noise. To compensate, we influence (6) by the standard deviation of X_t, which reflects energy and noisiness of the original signal, to obtain $b_j(x_t)$. A similar method is also reported in [15].

7 Stochastic Model of Time in Music Performance

As stated before, any model for timing synchronization of musical events should consider the hypothesis that the temporal structure of listeners' expectations is a dynamic structure. A primary function of such structures is *attentional*, which allows *anticipation* of future events, enabling perceptual targeting and coordination of action with musical events. These considerations led Large and Jones [9] to propose a model of meter perception, where they assume a small set of internal oscillations operating at periods that approximate those at hierarchical metrical levels. Each oscillator used in the model is *self-sustained* in the sense that, once activated, it can persist, even after simulation ceases or changes in significant ways. The oscillator has the ability to entrain to incoming rhythmic signals. Their model has been tested and verified largely in different experiments with human subjects. The tempo model introduced here is an adoption of the internal oscillator in [9] in a stochastic and continuous audio framework.

We define the problem as follows: given a music score such as the one in Fig. 1 with a global tempo as Ψ in *seconds/beat*,[4] depicting the (relative) beat duration of an event k by ℓ_k, the absolute clock-time event location in seconds can be obtained by the following recursive relationship:

$$T_k = T_{k-1} + \Psi \times \ell_k. \quad (7)$$

However, even if an entire piece is depicted with a fixed tempo, the tempo variable Ψ undergoes various dynamics and changes; it is from these changes that the expressivity of a musical performance is derived. Our goal here is to infer the dynamics of the tempo as a random variable through time.

7.1 Attentional Model of Tempo

Internal tempo is represented by a random variable Ψ_k revealing how fast the music is flowing with regard to the physical time. Following [9], we model the behavior of such

4. For example, $\Psi = 0.5$ seconds/beat for the sample score of Fig. 1 with tempo of 120 beats per minute.

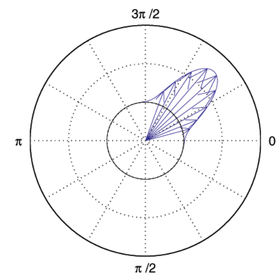

Fig. 7. Sample Von Mises distribution on a clockwise sine-circle map, with mean $7\pi/4$ or $-\pi/4$ and $\kappa = 15$.

a random variable as an internal oscillator entraining to the musician's performance. Such internal oscillations can be represented and modeled easily using sine-circle maps. These models have been well studied in the literature and can be considered as nonlinear models of oscillations that entrain to a periodic signal using discrete-time formalism. In this framework, the phase of the sine-circle map is an abstraction of time and corresponds to the time to pass one circular period or the local tempo. Using this framework, we represent the tempo random variable as ψ_k in seconds/beat and note onset positions as phase values ϕ_k on the sine circle. This way, given a local tempo ψ_i, the score onset time t_n can be represented as $\phi_n = \frac{t_n}{\psi_i} + 2k\pi$, where k is the number of tempo cycles to reach t_n. In our model, a phase advance is the portion of the oscillator's period corresponding to event *Interonset Intervals* (IOIs). Thus, if the tempo is assumed as fixed (ψ_k) throughout a piece, then

$$\phi_{n+1} = \phi_n + \frac{t_{n+1} - t_n}{\psi_k} \mod {}^{+\pi}_{-\pi} \quad (8)$$

would indicate relative phase position of events in the score.

In order to compensate for temporal fluctuations during live music performance, we would need a function of ϕ that would correct the phase during live synchronization, and at the same time, model the attentional effect discussed previously. The attentional pulse can be modeled using a periodic probability density function, the von Mises distribution, which is the circle map version of the Gaussian distribution, as depicted below:

$$f(\phi|\phi_\mu, \kappa) = \frac{1}{I_0} e^{\kappa \cos(2\pi(\phi - \phi_\mu))}, \quad (9)$$

where I_0 is a modified Bessel function of first kind and order zero, and ϕ_μ and κ are mean and variance equivalents of the von Mises distribution. Fig. 7 demonstrates a realization of this function on the sine-circle map.

It is shown in [9] that the corresponding phase coupling function (tempo correction factor) for this attentional pulse is the derivative of a unit amplitude version of the attentional function, as depicted in (10). Fig. 8 shows this function for different values of κ and $\phi_\mu = 0$:

$$F(\phi|\phi_\mu, \kappa) = \frac{1}{2\pi \exp \kappa} e^{\kappa \cos(2\pi(\phi - \phi_\mu))} \sin 2\pi(\phi - \phi_\mu). \quad (10)$$

With the above introduction, (8) can be rewritten as

$$\phi_{n+1} = \phi_n + \frac{t_{n+1} - t_n}{\psi_k} + \eta_\phi F(\phi_n|\phi_{\mu_n}, \kappa) \mod{}^{+\pi}_{-\pi}, \quad (11)$$

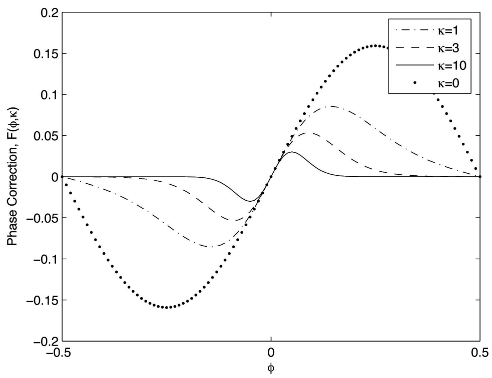

Fig. 8. Phase correction function in (10).

where η_ϕ is the coupling strength of the *phase coupling* equation and ϕ_{μ_n} is the *expected* phase position of the nth event in the score according to previous justifications.

Phase coupling is not sufficient in itself to model phase synchrony in the presence of a complex temporal fluctuation. To maintain synchrony, the period (or tempo) must also adapt in response to changes in sequence rate as follows:

$$\psi_{n+1} = \psi_n(1 + \eta_s F(\phi_n|\phi_{\mu_n}, \kappa)). \tag{12}$$

Equations (11) and (12) can recursively update tempo and expected onset positions upon onset arrivals of *temporal* events from the inference engine. However, note that the phase-time regions, where the phase adjustment is most efficient, in Fig. 8 are identical to the region around the mean of the attentional distribution (9) spanned by its variance κ. Smaller values of κ spread the correction over the phase domain, amounting to a wider *variance* in the attentional function meaning that expectancy is dispersed throughout the oscillator. For this reason, the parameter κ is usually referred to as *attentional focus*. This observation suggests that the values of κ should be adjusted at each update to obtain the best possible performance. To this end, before each tempo update, we solve for $\hat{\kappa}$ using a maximum-likelihood formulation on the dispersion about the mean of a sampled population of previously occurred ϕ_ns. This dispersion is given by the following equation on the circular map:

$$r = \frac{1}{n}\sum_{i=1}^{n} \cos 2\pi(\phi_i - \phi_{\mu_i}), \tag{13}$$

which can be easily calculated recursively in real time. Having this, the solution for $\hat{\kappa}$ is shown to be [24, Section 10.3.1]:

$$\hat{\kappa} = A_2^{-1}(r), \qquad \text{where} \qquad A_p(\lambda) = \frac{I_{p/2}(\lambda)}{I_{p/2-1}(\lambda)}, \tag{14}$$

where $I_\nu(\lambda)$ is the modified Bessel function of the first kind and order ν. The solution to $\hat{\kappa}$ in (14) is obtained by a table lookup of $A_2(\lambda)$ and using accumulated dispersions from (13) in real time.

7.2 Tempo Agent and Decoding

The tempo decoding scheme presented in this section is a recursive algorithm based on the above model and resembles an extended Kalman filtering approach [25]. The Kalman filter estimates a process by using a form of feedback control: The filter estimates the process state at some time and then obtains feedback in the form of environmental measurements. The general Kalman filter algorithm then falls within two steps: *Prediction* and *Correction*. The prediction equations are responsible for projecting forward (in time) the current state and error estimates to obtain the a priori estimates for the next time step. The correction equations are responsible for the feedback or incorporating the new measurement into the a priori estimate to obtain an improved a posteriori estimate. While general Kalman filters use linear estimators, Extended Kalman Filters (EKFs) assume nonlinear estimators (as in our case with the Mises-Von correction factors).

Algorithm 1 shows the two *correction* and *prediction* steps for the tempo agent. The correction procedures make use of the *true* arrival time of event n or t_n and within two steps: In the first, we update the true variance κ needed during updates by accumulating the circular dispersion as in (15) (a real-time approximation of (13)) by using an accumulation factor η_s that is set to a fixed value. Having κ updated through table lookup, the algorithm then updates the relative phase position of event n by using previous estimations, current measurements, and the score phase position. The prediction step then uses the newly corrected phase position of event n or ϕ_n, the score phase position $\hat{\phi}_n$, and the correction factors to obtain the new tempo prediction for event $n+1$. This algorithm is called recursively and upon each arrival of a newly aligned position from the audio agent.

Algorithm 1. Real-time Tempo decoding algorithm

Require: Upon decoding of event n at time t_n by the audio agent (measurement), given score IOI phase positions $\hat{\phi}_n$, initial or previously decoded tempo s_n

1: *Correction (1):* Update κ (variance)

$$r = r - \eta_s\left[r - \cos\left(2\pi\left(\frac{t_n - t_{n-1}}{\psi_k} - \hat{\phi}_n\right)\right)\right] \tag{15}$$

$$\kappa = A_2^{-1}(r) \quad \text{(Table lookup)}$$

2: *Correction (2):* Update ϕ_n

$$\phi_n = \phi_{n-1} + \frac{t_n - t_{n-1}}{\psi_{n-1}} + \eta_\phi F(\phi_{n-1}, \hat{\phi}_{n-1}, \kappa) \quad \text{mod}_{-\pi}^{+\pi}$$

3: *Prediction:*
$\psi_{n+1} = \psi_n[1 + \eta_s F(\phi_n, \hat{\phi}_n, \kappa)]$

4: **return** ψ_{n+1}

Due to the nature of the proposed model, the newly obtained tempo at each step ψ_n is a *predictive* tempo flow that can be used to anticipate future note locations in time. We use this feature in the next section to obtain the survival function needed for the inference module.

7.3 Survival Distribution Model

In Section 4, we introduced the global inference method used for a Hybrid Hidden Markov/semi-Markov model described in Section 3.1. We also introduced the graphical model topology with explicit time models with the use of explicit occupancy distributions $d_j(u)$, which are required to calculate the inference formulation in Section 4. In this section, we derive and justify our choice of occupancy distributions $d_j(u)$ needed during forward propagation.

We consider the process underlying the arrival rate of events over a time period of musical performance as a spatial Poisson process with distribution $P(N(t) = k)$, where $N(t)$ is the number of events that has occurred up to time t. The choice of this memoryless process is obviously an approximation and assumes that musical events arrive independently of each other. This process is characterized as

$$P[(N(t+\tau) - N(t)) = k] = \frac{e^{-\lambda(x,t)\tau}(\lambda(x,t)\tau)^k}{k!}, \qquad (16)$$

where $\lambda(x,t)$ is the expected number of events or arrivals that occurs at score location x and time t. We are now interested in a process that can model the arrival time of the kth event, or T_k, and from which we can derive the *survival function* needed for (3) and defined in (1). Depicting the real time as t and t_{n-1} as the previously decoded event, the survival distribution is

$$\begin{aligned} d_j(t - t_{n-1}) &= P(T_n > t | T_{n-1} = t_{n-1}, t_{n-1} < t) \\ &= P[(N(t_n) - N(t_{n-1})) = 0] \\ &= \exp[-\lambda(n,t)(t - t_{n-1})]. \end{aligned} \qquad (17)$$

Now that we have a direct formulation of the survival distribution, it only remains to specify $\lambda(n,t)$. Note that the expected value of this distribution is $1/\lambda$, which is, for event n, equivalent to its expected duration according to both the score and the latest tempo decoding as demonstrated in Section 7.1. Therefore,

$$\lambda(n,t) = \frac{1}{\psi_{n-1}\ell_n}, \qquad (18)$$

noting that s_n or the (real-time) decoded local tempo is a function of both time t and score location n. Combining both (18) and (17) provides us with the survival distribution to be used along with (3) during the inference process:

$$d_j(t - t_{n-1}) = \exp\left[-\frac{t - t_{n-1}}{\psi_{n-1}\ell_n}\right]. \qquad (19)$$

Note that the upper limit of the product u in (3) is also equal to the expected duration of the corresponding state or $\psi_j \ell_j$.

In summary, the tempo agent described in this section provides the occupancy function $d_j(u)$ as well as upper limits of (3) adaptively during a real-time performance, and decodes a continuous parameter pertaining to the tempo of the performance under consideration. Probably the most important characteristic of the proposed model is in its adaptability to a real-time context, thus modeling the dynamic temporal structure of music performance.

8 Evaluation

In this section, we provide the results of our real-time alignment method and temporal models, and evaluate them in various situations. The evaluation of score-following systems with regard to alignment was a topic in the MIREX2006 evaluation contest [26], [27]. In that contest, the organizers with the help of the research community prepared references over more than 45 minutes of concert acoustic music and defined certain evaluation criteria that we will reuse in this paper. However, no clear methodology has yet been proposed for the evaluation of tempo synchronization and timing accuracy, which is fundamentally a different topic than score alignment. In order to capture both, we propose two experimental setups for evaluation. In Section 8.1, we evaluate the timing accuracy of the system by artificially synthesizing temporal fluctuations and demonstrating different aspects of the adaptive decoding at work. In Section 8.2, we evaluate the system against real acoustic signals using the MIREX framework with some extensions.

Fig. 9. Sample score 1 for tempo experiment.

8.1 Evaluation of Dynamic Temporal Decoding

In this section, we evaluate the performance of the system against synthesized audio from a given score. The main reason for separating this procedure from real acoustic performances is for reassessment of the tempo synchronization and dynamic decoding of our temporal models. While evaluating alignment results is easily imaginable using real and acoustic data, the evaluation of tempo fluctuation is a difficult task. It is generally impossible to ask a musician to perform a given score using a temporal progression curve up to millisecond precision to be used as a reference. On the contrary, this is quite imaginable using synthesized audio by arranging temporal progressions of score events during the synthesis process.

Before defining the experimental setup and showing results, it is important to highlight several characteristics of the tempo agent described in Section 7 in terms of performance. First, the oscillator model has the underlying hypothesis that tempo progresses continuously and the tempo process adapts or locks into the new tempo progressively. This means that when an abrupt or discontinuous jump occurs in the tempo, κ or attentional focus should undergo abrupt changes until the tempo random variable reaches an equilibrium within a few steps. At the same time, when the tempo changes continuously (for example, in the case of an acceleration or deceleration), the agent should be capable of locking itself to the new tempi. We therefore study each case separately. In both experiments, we consider a simple score depicted in Fig. 9 containing 30 notes with a score (or prior) tempo of 60 bpm or 1 $\frac{\text{second}}{\text{beat}}$. By synthesizing this score to audio, we enforce a different tempo curve than the fixed tempo of the score and feed both the score and synthesized audio into the system and study the results.

The synthesis method used here is based on a simple FM synthesis method described in [28] and used in many commercial synthesizers. We did not experience any significant difference by changing the synthesis method regarding the aims and results for this section. Evaluation on more complex signals is discussed in Section 8.2.

8.1.1 *Discrete Tempo Jumps*

We first study the results and behavior of the system for discrete tempo jumps in the incoming audio. To this aim, we synthesize the score of Fig. 9 by introducing two tempo jumps during the life of the synthesized score of Fig. 9.

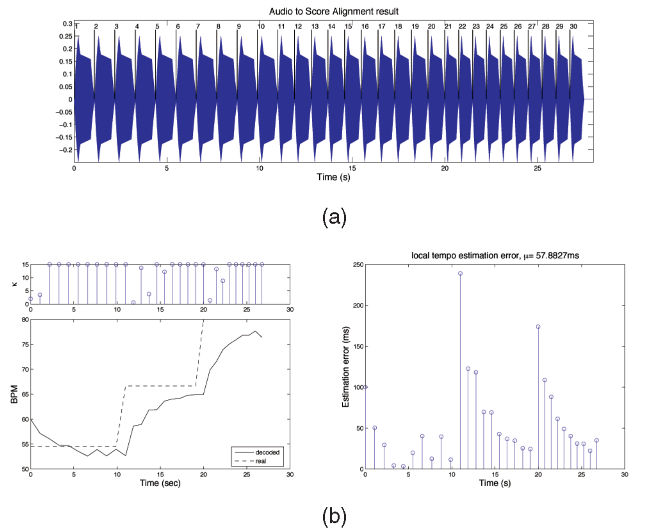

Fig. 10. Evaluation of tempo decoding using synthesized score and discretely controlled tempo. (a) Waveform and alignment result. (b) Estimated and real tempi for acceleration and deceleration in BPM.

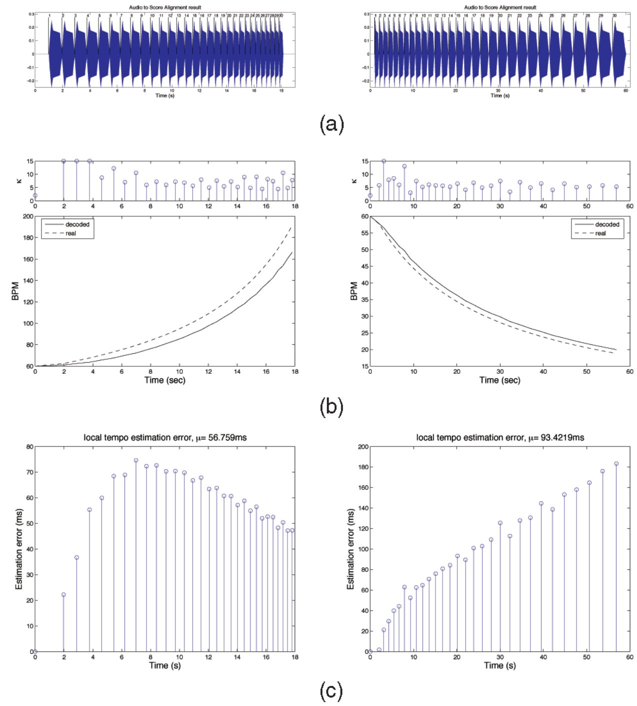

Fig. 11. Evaluation of tempo using synthesized score and continuously controlled tempo. (a) Waveforms and alignments for accelerating (left) and decelerating (right) tempi. (b) Estimated and real tempi for acceleration and deceleration in BPM. (c) Estimation error for acceleration (left) and deceleration (right).

Results are demonstrated in Fig. 10, where Fig. 10a shows the synthesized waveform with the alignment results where each number tag refers to one of the 30 notes in the score in Fig. 9. Comparing the left and right portions of this waveform clearly shows the difference in duration length of each event corresponding to the abrupt tempo jump. Fig. 10b shows the tempo synchronization result along with the real tempo curve as a dashed line on the main left figure and the corresponding κ parameter at each time step on the top, and local estimation error on the right figure. The estimation error is computed as the difference in millisecond between the real tempo and decoded tempo both expressed in milliseconds/beat.

Looking closely at Fig. 10, we can interpret the online tempo adaptation as follows: Within the three regions where the reference tempo is different from the expected tempo, the agent goes into sudden instability, leading to the biggest estimation error, as depicted in Fig. 10b on the right. These instabilities lead to sudden changes in the κ parameter, which controls attentional focus. This process continues for several time steps until the agent locks itself around the correct tempo that can be observed by looking at estimated tempi converging to the reference tempi, or by observing the decrease in the estimation error, as well as by observing the increase in the adaptive κ parameter reaching its upper bound (here set to 15). Note also that the reference tempo curve for audio synthesis starts with a different tempo than the prior one indicated by the score, so the κ parameter starts low in the beginning until stability and undergoes change every time the system enters in equilibrium as shown in Fig. 10b. The mean estimation error for this test session is 57 ms.

8.1.2 Continuous Tempo Change

For this experiment, we use the same procedure as before, but continuously change the tempo parameter during synthesis. This experiment is aimed at simulating acceleration and deceleration common in music performance practice. The control function for tempo during sound synthesis is set to an exponential function $e^{\gamma(n-1)}$, where n is the note event number in the score and γ controls the slope of the change with $\gamma < 0$ indicating acceleration and $\gamma > 0$ deceleration over performance time. A partial goal here is to demonstrate the performance of the system despite the lack of time to reach an equilibrium state.

Before doing a mass evaluation, we visually demonstrate some results to highlight the performance of the system. Fig. 11 shows the output of synchronization on acceleration (left) and deceleration (right) with $\gamma = \mp 0.04$ resulting in a tempo difference of 131 bpm and $-41\ \mathrm{bpm}$, respectively. As before, we are demonstrating the resulting synthesis waveforms and alignment tags in Fig. 11a, the real and estimated tempi along with adaptive κ parameters in Fig. 11b, as well as tempo estimation error on each event in Fig. 11c.

Fig. 11 leads to the following important observations: First, the κ parameter is constantly changing over the course of both processes in Fig. 11b. This is normal since the reference tempo is continuously evolving in both cases. Second, note that, while γ only changes signs in the two cases, the estimation results and the mean errors are quite different. This phenomenon is easy to explain: In the deceleration case (right portion in Fig. 11), the difference between the two tempo extremes is about $-40\ \mathrm{bpm}$ but the time steps between each event (and their respective tempo-phase) are exponentially increasing, so the system needs more time and steps to reach a better stability point, despite it following the original curve correctly. This leads to a bigger estimation error than the acceleration case, where the phase steps become smaller and smaller at each step. This observation is further enhanced by noticing that the estimation error for the acceleration curve (left of Fig. 11c) decreases after a while, in contrast to the deceleration case.

TABLE 1
Batch Results over Different Exponential Tempo Curves

γ	Length (s)	ΔS (bpm)	**Tempo Err (ms)**	**Onset Err (ms)**
-0.06	16.0	-49.5	68.22	9.50
-0.05	17.0	-46.0	62.51	9.35
-0.04	19.0	-41.0	56.32	9.73
-0.03	22.0	-34.9	44.02	9.27
-0.02	24.0	-26.4	37.87	9.26
0.02	43.0	47.1	8.13	10.82
0.03	51.0	83.2	44.44	10.34
0.04	61.0	131.4	93.46	9.59
0.05	73.0	195.7	104.68	9.50
0.06	88.0	281.8	158.78	8.69

The observations above are further enhanced by enlarging the evaluation set by varying the values of γ during sound synthesis. Table 1 shows the same evaluation procedure above for various values of γ, where the first three columns characterize the synthesized audio from score in Fig. 9, and the last two columns show tempo and onset estimation errors in milliseconds. Here again we can observe that accelerating $|\gamma|$s (or $\gamma > 0$) have better estimation rates than their decelerating counterparts. The estimation errors here are the mean over all of the events in the score (total of 30 in each case). The reader might argue that an estimated error of 158 ms (reported in the last row of Table 1) is not acceptable for a tempo synchronization application. In response, note that the tempo difference for this process (281.8 bpm) is almost never experienced in a musical performance setting unless explicitly stated in the music score by a discrete tempo change, which would resolve the case.

8.2 Evaluation of Alignment Precision

In Table 1, we report the mean onset error, which is the elapsed time between the detected time of each event and the synthesis reference. While these results are encouraging, in a real acoustic music situation, the audio signals are much less stationary than the synthesized signals used in the previous section. In this section, we evaluate the real-time alignment results in the context of acoustic music performances.

In 2006, an international evaluation campaign was organized by the research community for the evaluation of audio to score alignment algorithm for Music Information Retrieval Evaluation eXchange (MIREX) and was reported during the ISMIR conference in Victoria, Canada, in August 2006. The campaign was repeated in 2008 with more results and participants. During this campaign, a general consensus was obtained for evaluation metrics and procedures applicable to most available systems. The agreed procedures as well as documentation of all details and discussions are available through the MIREX Web-portal [26] and in [27]. Evaluation consists of running the system on a database of real audio performances with their music scores, where an alignment reference exists for each audio/score couple. This procedure aims at simulating a real-time performance situation; thus audio frames are required to enter incrementally into the system but the procedure could also be easily extended to non-real-time techniques.

Table 2 describes the database used for this evaluation, which is a partial copy of the one in [26] plus some additions. Items 1 and 2 are strictly monophonic, item 3 is lightly polyphonic with the appearances of music chords of the violin from time to time in the piece, while item 4 is strictly polyphonic with up to four different voices happening at the same time. This database contains more than 30 minutes of referenced audio/score pairs and has been chosen to demonstrate the performance of the system on different musical instruments, and styles (item 1 is in contemporary music style with unconventional timings) and degree of polyphony. Items 1-3 are used in [26], whereas item 4 is aligned using a heavy offline algorithm and further enhanced as reported in [29]. All of the symbolic scores in this database contain some forms of *special timed* events of Section 5.2 (e.g., musical trills), which are detected automatically in the original score by our system's score parser and prepared for performance.

TABLE 2
Evaluation Database Description

#	**Piece name**	**Composer**	**Instr.**	**Files**	**Prefix**	**Events**
1	Explosante-Fixe	P. Boulez	Flute	7	*tx-sy*	615
2	K. 370	Mozart	Clarinet	2	k370	1816
3	Violin Sonata 1	J.S. Bach	Violin	2	vs1-	2019
4	Fugue BWV.847	J.S. Bach	Piano	1	RA	225

Once every piece is run through the system, we obtain a set of event tags i with their detection times t_i^d in milliseconds. The process of evaluation compares the results with the previously prepared references for each piece with the same tags i and alignment times t_i^r. An event i is reported as *missing* if either t_i^d does not exist or $|t_i^d - t_i^r| > 250$ ms, meaning that the error tolerance for matching is set to 250 milliseconds. Evaluation metrics are then the number of misses, and corresponding statistics on the offset time $o_i = t_i^d - t_i^r$ between detected time tags and the associated ones in the reference database. Table 3 shows the results of evaluation on each file in the described database, starting from monophonic scores and going gradually toward the polyphonic ones. Here, *FP* refers to *false positives*, which are misaligned events and are parts of the *missed* events. The *average offset* error is the mean over the absolute offset values or $\sum |o_i|$, where *mean offset* is the regular mean without taking the absolute value. Given these statistics, the *overall precision* is calculated as the percentage of total number of events to detect minus the total number of missed notes,

TABLE 3
Real-Time Alignment Evaluation Results

Source Info		Offset (ms)			Percentage	
Filename	Events	Average	Mean	STD	Missed	FP
K370.030	908	188.4	188.4	255.3	7.49%	0.22%
K370.032	908	166.1	166.1	208.9	5.95%	0.22%
s01	88	85.7	85.7	24.8	2.27%	0.00%
s04	76	81.7	81.7	29.0	5.26%	0.00%
s06	108	75.1	75.1	34.6	4.63%	0.00%
s11	63	109.4	109.4	217.4	17.46%	0.00%
t7-s03	90	115.3	115.3	63.9	6.67%	0.00%
t7-s16	98	113.0	113.0	26.2	5.10%	0.00%
t7-s21	92	106.0	106.0	25.4	3.26%	0.00%
vs1-4prs	1604	240.9	240.9	165.0	10.41%	0.00%
vs1-1ada	415	130.1	130.1	106.6	12.53%	1.45%
RA-C025D	225	99.8	99.8	75.3	9.33%	0.00%
Total Precision:		**91.49%**				
Piecewise Precision:		**92.47%**				

TABLE 4
Real-Time Alignment Comparative Evaluation

	Proposed	HMM	DTW	Pitch-based
Piecewise Precision:	91.5%	85.7%	51.2%	55.6%

whereas the *piecewise precision* is the mean of the same rate but over individual files. In [26], another metric is proposed pertaining to *latency* and defined as the interval between the detection time and the time the event is reported. This metric was specifically designed for systems that are *real time* but are not necessarily *online*, thus allowing a delay in the reporting of the correct alignment. This is the case, for example, in [15]. We drop this measure since our system is strictly online and this measure is always zero.

For the sake of completeness, we provide a comparison of piecewise precision performance of our proposed model to other existing approaches in Table 4. System identifiers correspond to the following designs: *HMM* system corresponds to a pure HMM solution to the problem as presented in [13] and expanded further in [30]. *DTW* refers to a solution using online Dynamic Time Warping on chroma features loosely based on [31] and [32], and *Pitch-based* refers to a string-matching type algorithm based on a pitch detector input as reported in [10]. Shown results for *HMM* and *Pitch-Based* categories are borrowed from results on the same database in [26]. Reported results are based on a subset of the database in Table 2, excluding the highly polyphonic item 4. Moreover, the described evaluation scheme requires high alignment precision up to a few milliseconds, which might not be a priority for many of the mentioned approaches. The reader curious about the topic of score following evaluation is encouraged to check the MIREX Web portal,[5] where every year, new systems and approaches are being tested and evaluated against each other.

Despite the adopted harsh evaluation scheme, the robustness of the proposed model in highly polyphonic situations pinpoints to an important design parameter: in a situation where observations on data streams are uncertain, coupling different sources of information and tackling adaptive time models instead of adopting approximate schemes could result in less uncertainty than observed in each individual agent.

9 Conclusion

In this paper, we presented the design and implementation of a real-time and online audio to score alignment for music signals. The presented design features a coupled tempo/audio inference model that adaptively updates its explicit duration models during performance, and hence, does not need any offline training or parameter tweaking. The system is capable of decoding the placement of the live musician in the score and also provides a continuous tempo parameter that is quite useful for automatic accompaniment applications.

Real-time music information retrieval poses interesting challenges to the engineering community due to the natural complexity of musical structures, to the extent that classical approaches to speech processing prove to be insufficient to address all the complexity of the music-related issues. The methods presented in this paper try to tackle two main issues that are usually passed through approximations in speech processing: duration-specific models that are adaptive to the real-time context, and coupling different sources of information with uncertainty through a unique inference technique to increase precision. The proposed anticipatory framework emerged out of considerations in tackling the specific musical question of real-time synchronization, but the methods presented can be extended to other sequential applications that require the given premises. We believe that music, as a rich source of complexity that is unforgiving to approximation, could inspire more advances in machine intelligence.

The system presented in this paper has been purposely designed and developed to be used in serious concert situations that go beyond formal evaluations. To date, it has been used in several large-scale music events worldwide, including performances with the Los Angeles Philharmonic and in Japan, France, Germany, and with more performances scheduled for seasons to come. We believe that, in the end, the best evaluation of any real-time and performance-oriented system is by its usability by the community. We invite curious readers to follow events featuring our system, watch demos, or download and test the system themselves.[6]

Appendix

Derivation of Forward Recursion

To solve for an analytical solution of the inference formulation, we are interested in the most likely state sequence S_0^τ that would generate the outside process X_0^τ up to time τ and over the entire state-space $\{s_0, \ldots, s_J\}$. By definition and applying the Bayes formula in chains, we have:

$$\begin{aligned}\alpha_j(t) &= \max_{s_0,\ldots,s_{t-1}} P\big(S_{t+1} \neq j, S_t = j, S_0^{t-1} = s_0^{t-1}, X_0^t = x_0^t\big)\\ &= \max_{1\le u\le t}\max_{i\neq j} P\big(S_{t+1} \neq j, S_{t-u} = j, v = 0, \ldots, u-1,\\ &\qquad S_{t-u} = i | X_0^t = x_0^t\big)\\ &= \max_{1\le u\le t} \frac{P\big(X_{t-u+1}^t = x_{t-u+1}^t | S_{t-v} = j, v = 0, \ldots, u-1\big)}{P\big(X_{t-u+1}^t = x_{t-u+1}^t | X_0^{t-u} = x_0^{t-u}\big)},\end{aligned} \tag{20}$$

$$\times P(S_{t+1} \neq j, S_{t-v} = j, v = 0, \ldots, u-2 | S_{t-u+1} = j, S_{t-u} \neq j), \tag{21}$$

$$\times \max_{i\neq j} P(S_{t-u+1} = j | S_{t-u+1} \neq i, S_{t-u} = i), \tag{22}$$

$$\times P\big(S_{t-u+1} \neq i, S_{t-u} = i | X_0^{t-u} = x_0^{t-u}\big). \tag{23}$$

The nominator in (20) reduces to $\prod_{v=1}^{u-1} b_j(x_{t-v})$ with the assumption that observations b_j are independent. The denominator here is a normalization factor that can be dropped out in our computation. Equation (21) is the definition of the occupancy distribution $d_j(u)$ from

5. http://www.music-ir.org/mirex/2009/index.php/Main_Page.

6. http://imtr.ircam.fr/index.php/Antescofo.

Section 3.1. Similarly, (23) is the definition of the semi-Markovian transition probabilities p_{ij} and (22) is the definition of α_i at time $t-u$. Replacing these definitions in the equation and factoring indexes, the recursion then becomes

$$\begin{aligned}\alpha_j(t) &= \max_{s_0,\dots,s_{t-1}} P\big(S_{t+1} \neq j, S_t = j, S_0^{t-1} = s_0^{t-1}, X_0^t = x_0^t\big)\\ &= b_j(x_t)\\ &\times \max\left[\max_{1\le u\le t}\left(\left\{\prod_{v=1}^{u-1} b_j(x_{t-v})\right\} d_j(u) \max_{i\neq j}(p_{ij}\alpha_i(t-u))\right)\right].\end{aligned} \tag{24}$$

Acknowledgments

This research was partially funded by the European Commission in the framework of the FP7 SAME project. The author would like to specially thank composer Marco Stroppa, who initiated the artistic challenges that led to the design exposed in this paper. The system descibed in this paper, `Antescofo`, has been designed specifically for a new work by Stroppa and is today in use for other music pieces and more new works to come.

References

[1] R.B. Dannenberg, "An On-Line Algorithm for Real-Time Accompaniment," *Proc. Int'l Computer Music Conf.*, pp. 193-198, 1984.

[2] B. Vercoe, "The Synthetic Performer in the Context of Live Performance," *Proc. Int'l Computer Music Conf.*, pp. 199-200, 1984.

[3] R.J. McNab, L.A. Smith, I.H. Witten, C.L. Henderson, and S.J. Cunningham, "Towards the Digital Music Library: Tune Retrieval from Acoustic Input," *Proc. First ACM Int'l Conf. Digital Libraries*, pp. 11-18, 1996.

[4] R.B. Dannenberg, "An Intelligent Multi-Track Audio Editor," *Proc. Int'l Computer Music Conf.*, vol. 2, pp. 89-94, Aug. 2007.

[5] J. Keshet, S. Shalev-Shwartz, Y. Singer, and D. Chazan, "A Large Margin Algorithm for Speech-to-Phoneme and Music-to-Score Alignment," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 15, no. 8, pp. 2373-2382, Nov. 2007.

[6] M. Johnson, "Capacity and Complexity of HMM Duration Modeling Techniques," *IEEE Signal Processing Letters*, vol. 12, no. 5, pp. 407-410, May 2005.

[7] L. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257-285, Feb. 1989.

[8] R. Rosen, *Anticipatory Systems*, vol. 1. Pergamon Press, 1985.

[9] E.W. Large and M.R. Jones, "Dynamics of Attending: How People Track Time-Varying Events," *Psychological Rev.*, vol. 106, no. 1, pp. 119-159, 1999.

[10] M. Puckette and C. Lippe, "Score Following in Practice," *Proc. Int'l Computer Music Conf.*, pp. 182-185, 1992.

[11] L. Grubb and R.B. Dannenberg, "A Stochastic Method of Tracking a Vocal Performer," *Proc. Int'l Computer Music Conf.*, pp. 301-308, 1997.

[12] C. Raphael, "Automatic Segmentation of Acoustic Musical Signals Using Hidden Markov Models," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 21, no. 4, pp. 360-370, Apr. 1999.

[13] N. Orio and F. Déchelle, "Score Following Using Spectral Analysis and Hidden Markov Models," *Proc. Int'l Computer Music Conf.*, 2001.

[14] A. Cont, "Realtime Audio to Score Alignment for Polyphonic Music Instruments Using Sparse Non-Negative Constraints and Hierarchical HMMS," *Proc. IEEE Int'l Conf. Acoustics, Speech, and Signal Processing*, May 2006.

[15] C. Raphael, "Aligning Music Audio with Symbolic Scores Using a Hybrid Graphical Model," *Machine Learning*, vol. 65, nos. 2/3, pp. 389-409, 2006, http://dx.doi.org/10.1007/s10994-006-8415-3.

[16] M. Müller, *Information Retrieval for Music and Motion*. Springer-Verlag New York, Inc., 2007.

[17] P. Boulez, *Penser la Musique Aujourd'hui*. Gallimard, 1964.

[18] I. Xenakis, *Formalized Music*. Univ. of Indiana Press, 1971.

[19] J.D. Ferguson, "Variable Duration Models for Speech," *Proc. Symp. Applications of Hidden Markov Models to Text and Speech*, pp. 143-179, Oct. 1980.

[20] A. Cont, "Antescofo: Anticipatory Synchronization and Control of Interactive Parameters in Computer Music," *Proc. Int'l Computer Music Conf.*, Aug. 2008.

[21] I. Peretz and R.J. Zatorre, "Brain Organization for Music Processing," *Ann. Rev. Psychology*, vol. 56, pp. 89-114, 2005.

[22] Y. Guédon, "Hidden Hybrid Markov/Semi-Markov Chains," *Computational Statistics and Data Analysis*, vol. 49, pp. 663-688, 2005.

[23] K.P. Murphy, "Dynamic Bayesian Networks: Representation, Inference and Learning," PhD dissertation, Computer Science Division, Univ. of California Berkeley, http://www.cs.ubc.ca/murphyk/Thesis/thesis.html, 2002.

[24] K.V. Mardia and P. Jupp, *Directional Statistics*, second ed. John Wiley and Sons Ltd., 2000.

[25] P.S. Maybeck, *Stochastic Models, Estimation and Control*, vol. 1. Academic Press, 1979.

[26] ScofoMIREX, "Score Following Evaluation Proposal," http://www.music-ir.org/mirex/2006/index.php/Score_Following_Proposal, Aug. 2006.

[27] A. Cont, D. Schwarz, N. Schnell, and C. Raphael, "Evaluation of Real-Time Audio-to-Score Alignment," *Proc. Int'l Symp. Music Information Retrieval*, Oct. 2007.

[28] F.R. Moore, *Elements of Computer Music*. Prentice-Hall, Inc., 1990.

[29] C. Yeh, N. Bogaards, and A. Roebel, "Synthesized Polyphonic Music Database with Verifiable Ground Truth for Multiple f0 Estimation" *Proc. Eighth Int'l Conf. Music Information Retrieval*, pp. 393-398, Sept. 2007.

[30] A. Cont, D. Schwarz, and N. Schnell, "Training Ircam's Score Follower," *Proc. IEEE Int'l Conf. Acoustics, Speech, and Signal Processing*, Mar. 2005.

[31] S. Dixon, "Live Tracking of Musical Performances Using On-Line Time Warping," *Proc. Eighth Int'l Conf. Digital Audio Effects*, 2005.

[32] R.B. Dannenberg and N. Hu, "Polyphonic Audio Matching for Score Following and Intelligent Audio Editors," *Proc. Int'l Computer Music Conf.*, pp. 27-34, 2003.

Arshia Cont received the BS degrees in electrical engineering and applied mathematics from Virginia Tech, the master's degree in acoustics, signal processing, and computer science applied to music from the University of Paris 6, and the joint PhD degree from the University of California at San Diego and the Ircam-Centre Pompidou in Paris. He is currently a researcher at the Ircam-Centre Pompidou and also serves as the scientific liaison on several electronic arts projects at the Ircam-Centre Pompidou in Paris with premieres in concert halls worldwide featuring applications of his research on real-time music processing systems. He is a student member of the IEEE.