Musical Accompaniment Systems

Christopher Raphael
Dept. of Mathematics and Statistics
Univ. of Massachusetts, Amherst*

March 14, 2004

In a futuristic conception of music-making in the 24th century, *Star Trek: The Next Generation* captain Jean-Luc Picard plays a flute solo accompanied by a computer. The computer actually seems to hear the captain, and responds to the subtle nuances of his playing as a human accompanist would. This fantasy has been shared by a great many would-be soloists who long to take their rightful places on the stage. Unfortunately, while many have the necessary musical soul, almost by definition, only a very few will ever experience the dream of appearing as a soloist accompanied by a large ensemble of musicians.. In this way the need expressed in the Star Trek scene is acutely felt by many of today's musicians; fortunately, this dream is on the verge of becoming a reality.

Musical accompaniment systems trace their roots back to the familiar "Music Minus One"(MMO) recordings. MMO makes a recording of a piece of music for soloist and accompaniment, such as a sonata or concerto, in which only the accompaniment is actually recorded. A player of the featured instrument would then, instrument in hand, play the recording while attempting to perform the missing solo part. A heartfelt yet futile battle of wills follows which eventually results in the live player's unconditional surrender to the robotic insistence of the recording. Thus, contrary to both musical etiquette and common sense, the soloist must follow the accompaniment.

While one shouldn't underestimate the pedagogical and entertainment value of these recordings, most serious musicians view MMO as the antithesis of the musical ideal: the accompaniment should follow the soloist, and not the other way around. Musical accompaniment embrace the same essential goal of MMO — replicating the experience of a musical soloist, but rebel against the stifling reality of MMO. These systems are computer programs that actually *listen* to the live player while generating a flexible accompaniment *follows* the musician. Accompaniment systems can, through a series of rehearsals, even *learn* from the live musician to better anticipate and follow future performances.

Musical accompaniment systems made their debut at the 1984 International Computer Music Conference, held at the famous *Institut de Recherche et Coordination Acoustique/Musique* (IRCAM) in Paris, France. As a rather strange coincidence, two such systems were demonstrated, one by Roger Dannenberg [1], and one by Barry Vercoe [2]. While subsequent improvements in microprocessor capabilities and audio hardware paved the way for later efforts, these systems were built from "whole cloth" by cobbling together incongruous collections of hardware components. The more successful demonstration, given by Dannenberg, was based on a computer system designed specifically for the accompaniment problem. Using the then-current 6502 processor (of Apple II fame), Dannenberg wired together circuitry to accomplish the pitch extraction and sound synthesis tasks. The actual system, used to accompany Dannenberg's own trumpet playing, was built to satisfy the carry-on requirement for the trip to Paris and can be seen at http://fafner.math.umass.edu/chance. The demonstration made a lasting impression on the computer music community and set the stage for future progress. Vercoe's approach was based on a much larger PDP-11

*This work supported by NSF grant IIS-0113496.

Figure 1: Opening phrase of B. Marcello's concerto for oboe and strings

computer requiring the performance to be piped in from a different room at IRCAM using closed-circuit television. Vercoe's methods used a combination of sensors mounted on the keys of a flute, and audio analysis based on current DSP hardware to process the incoming sound signal. Unfortunately, the system got lost in performance and, after a period of confusion, further demonstration was postponed.

These systems clearly demonstrated the potential and potential pitfalls of this intriguing musical application, showing that the task was possible while leaving the problem wide open for future research. We proceed with a look "under the hood" of the musical accompaniment system, and in doing so break the problem down into two basic subtasks, "hearing" and "playing."

1 Hearing the Soloist

Before a computer (or human) can accompany a soloist, it must first *hear* the soloist. Some approaches to the musical accompaniment problem circumvent this obstacle by either using MIDI (Musical Instrument Digital Interface) input, which requires the live player to play from an electronic instrument, or using special hardware to simplify the problem, such as sensors mounted on the instrument or pitch-to-MIDI converters. These approaches have significant disadvantages: Requiring MIDI input, as in [3], is unattractive to musicians who play acoustic instruments while the use of sensors is cumbersome and requires specialized hardware not generally available. Moreover, all of these methods essentially constitute feature extraction techniques, only delaying the heart of the hearing problem. We will instead focus on machine listening for the actual audio signal. While appropriate for any kind of music, we believe audio analysis the natural point of departure for classical or vocal music where it is generally the only easily accessible data from which the live player's behavior can be inferred. Thus, the raw data for this problem are the samples composing the acoustic signal generated by the live player.

Much music is composed predominantly of pitched sounds which, from a physical point of view, have nearly periodic waveforms. For this reason time/frequency representations, such as the *spectrogram*, can be particularly illuminating. Figure 1 shows a spectrogram of a particularly simple excerpt from the Adagio of the Marcello oboe concerto. This figure was generated by partitioning the audio signal into 30 ms. segments or "frames," while drawing the power spectrum (squared modulus of the finite Fourier transform) of each frame in consecutive columns. Thus, the spectrogram shows frequency content evolving over time. In the figure, each note of the oboe generates bands of energy at integral multiples of the fundamental frequency, corresponding the the harmonics or overtones. This is consistent with the familiar result from Fourier analysis that represents a periodic functions as a sum of sines and cosines with integrally related periods. Figure 2 shows a more typical example from the soloist's entrance in Brahms' violin concerto. While the essential signature of each note is as before, the listening problem is complicated by many factors such as fast notes, vibrato, discrepancies in pitch, significant signal contribution from the accompanying instruments, variable timbre on the part of the soloist, as well as the inevitable errors humans make.

The automatic transcription of an acoustic signal as in Figure 2 would be challenging to say the least. Luckily, our hearing problem is greatly simplified by our prior knowledge of the musical score, giving

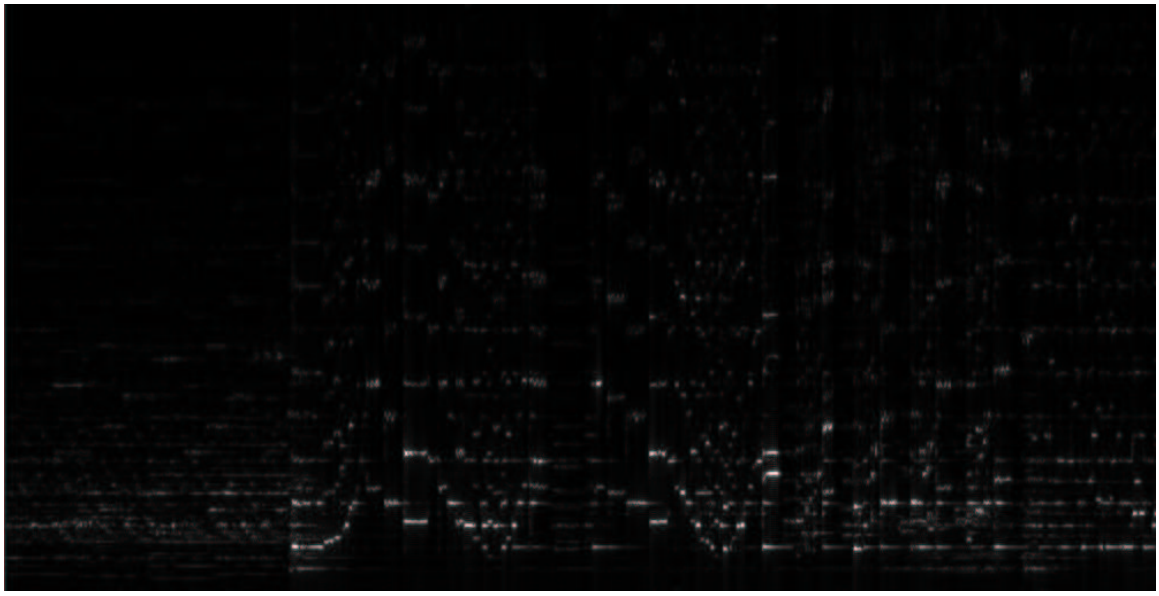


Figure 2: Soloist's entrance in Brahms' violin concerto

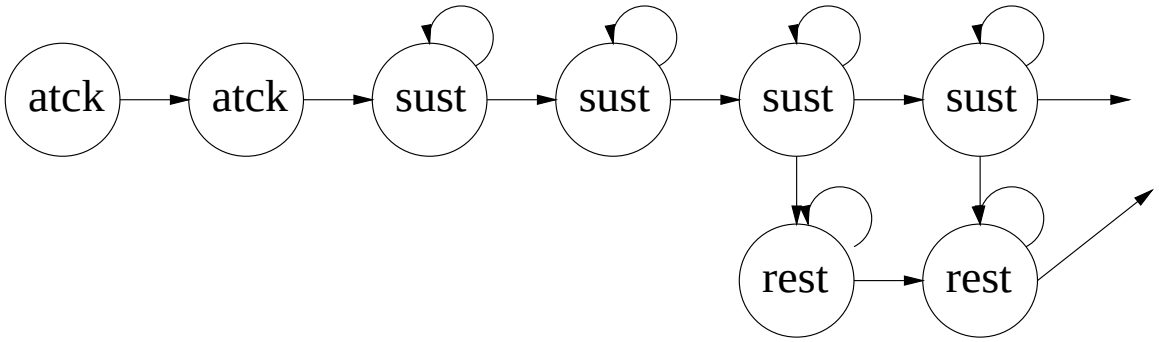
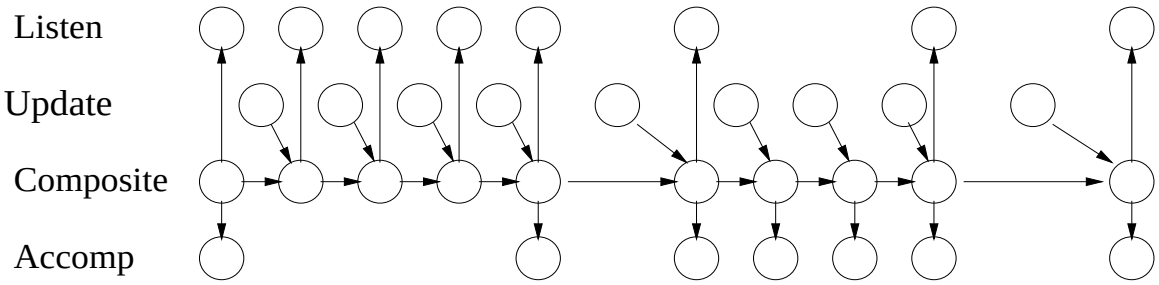


Figure 3: A Markov model for a note

the sequence of pitches the soloist will play, as well as their approximate durations. Thus, the listening problem can be cast as a search for the optimal warping of the score to match the actual audio data. Most recent work on this problem employs this vantage point and seeks an optimal warping by using dynamic programming in one way or another [6], [4], [5], [7], [8]. After this common point of departure, research approaches diverge quickly. While the role of statistics in this problem is still being debated, we present a statistical approach (a hidden Markov model) for score following here. The results given by HMMs compare quite favorably with competing methods in this domain, are likely to resonate with this readership, and are gaining currency among researchers in score following due to their performance, computational tractability, and automatic adaptability.

Consider the Markov chain note model of Figure 3. In the model each state moves probabilistically to one of its possible successors, shown by the arrows in the figure, with no regard for the past state history. The HMM assumes that every time we visit a state we generate a "frame" of data (column of spectrogram) from some distribution characteristic of that state. For instance, during the sounding portion of the note the distribution would favor spectra with the characteristic bands of energy associated with the current pitch in the musical score. This particular model contains several "attack" states which attempt capture the characteristic and somewhat chaotic behavior at the start of a note, "sustain" states for the steady



state pitched part of the note, as well as an option for silence before the next note begins (e. g. as in a staccato note). The self-loops allow for variation in the number of frames associated with a particular note. The number of states and transition probabilities can then be chosen to achieve some desired note length distribution. Our hidden Markov chain is then constructed by creating a model like that of Figure 3 for each note in the score and linking them together to form a large directed graph composed of thousands of states.

Once the sound data are observed, the HMM can be used to make inferences about the hidden state process, thereby "hearing" the soloist. For instance, in the "on-line" (real-time analysis) problem, we wish to detect the onset times of the notes of the score with as little latency as possible. A given note can be detected by gathering frame data and, continually recomputing the probability that the hidden state variable is beyond the start state of the given note. When this probability exceeds some threshold, we assume the note is in the past and use all current data to compute the most likely frame to occupy the note onset. This frame will be our estimated note onset time. The "off-line" problem begins with a complete performance and seeks a warping between the score and sound data. The web page http://fafner.math.umass.edu/chance contains an audio file of the Brahms example with clicks superimposed over the most likely onset times for each note in the score.

2 Planning the Accompaniment

As the soloist is heard, the accompanist must construct the other musical parts around this incoming stream of observations. One of the primary difficulties here is the inevitable latency in our detection of solo notes: if we attempt to synchronize an accompaniment note with a solo note by waiting until the soloist's note is heard and then playing, the accompaniment will always be late. The human musician understands this hearing latency as part of the basic problem of ensemble playing and responds by continually predicting the future musical evolution based on what has been heard at any given moment. One way or another, the computer accompanist must grapple with this fundamental problem as well. We describe here an approach based on a statistical model for musical evolution. Other approaches estimate a running tempo and use this as the basis for predictions ([6], [2]).

As with the human accompanist, computerized musical accompaniment requires the fusion of several disparate knowledge sources. One knowledge source is, of course, the result of hearing process, which, for us, is a running commentary on the soloist's performance identifying the onset time of each note, delivered with variable latency. In addition, the accompanist must also understand the basic template for musical performance described in the musical score, (notes, rhythms, etc.), thereby allowing one to "sight-read" (perform with no training) credibly. However, the computer accompanist must also be able to improve over successive rehearsals much as live musicians do; thus another knowledge source we use consists of a collection of past solo performances, demonstrating the soloist's rhythmic interpretation. Finally, several performances of the accompaniment by human musicians allow us to capture a sense of musicality in places where it cannot be inferred from the soloist.

We model the problem through a collection of hundreds of Gaussian random variables whose mutual dependence is expressed through a graph — a Bayesian Belief Network. The backbone of the network is a model for musical evolution on the *composite* rhythm of both musical parts together:

$$\begin{array}{rcl} t_{n+1} &=& t_n + \tau_n \\ s_{n+1} &=& s_n + l_n \times t_n + \sigma_n \end{array}$$

where t_n is the time of the nth musical event, s_n is the local tempo in seconds per measure, l_n is the length, in measures, of the nth musical event, and $\{(\tau_n, \sigma_n)\}$ are independent normally distributed random vectors. Without the $\{(\tau_n, \sigma_n)\}$ variables, the model would produce a robotic rhythmic interpretation of the music, however the inclusion of these variables allows us to capture a rich variety of rhythmic interpretations. Note that the σ variables represent changes in tempo, while the τ variables represent note-by-note elongations or compressions without any tempo change (as in an *agogic* accent). It is through the *distributions* of these vectors that we represent a rhythmic interpretation, both in terms of average performance (mean vector) and repeatability (covariance matrices). The middle two layers of figure 3 provide a graphical description of the above model where the layer labeled as "Composite" is composed of the (t_n, s_n) vectors while the "Update" layer is composed of the (τ_n, σ_n) vectors. In interpreting this picture, the variables at each node in the graph depend only on their "parents" given all variables upstream.

Where does our hearing of the soloist fit in? We model the detected solo onset times as noisy observations of the true note onset times $\{t_n\}$, as in the top row of the figure. A similar treatment of the accompaniment onset times completes the picture of figure 3.

The problem of learning from rehearsals has a simple analog within the context of our model. When a section of music is rehearsed, an off-line analysis of the acoustic signal results in a sequence of estimated onset times as discussed in the "Hearing" section. Several such rehearsal lead to several sequences which can be collectively used in a parameter estimation problem. We seek the configuration of means and covariances of the $\{(\tau_n, \sigma_n\}$ variables that best explains the observed data — a maximum likelihood estimate. Human performances of the accompaniment part can be used similarly to train the musicality of the accompanist in situations where the accompanist must lead. The actual training process is a standard application of the EM algorithm.

The accompaniment task now reduces to the problem of scheduling the pending accompaniment note. At an point in time, some solo note onsets will have been detected and some accompaniment notes played. These variables are treated as observed in our model, and we use standard ideas from Bayesian Belief networks to compute the conditional distribution of the onset of the pending accompaniment note. It is often the case that new solo notes will be detected before the pending accompaniment note is actually played. In this case, the onset distribution for the pending accompaniment note is recomputed and the note is rescheduled using the new information. Thus an accompaniment can be rescheduled several times before it is actually played, while the eventual onset time makes use of *all* currently available information. The actual time that an accompaniment note is actually played will be influenced by all knowledge sources mentioned earlier: the score, the on-line analysis of the soloist's playing, the interpretation demonstrated by the soloist in rehearsal, as well as any human performances of the accompaniment that are available.

3 Producing the Output

The hearing component and the note scheduling mechanism account for the lion's share of the accompaniment task, however the end result must, of course, be sound. The simplest way to accomplish this is through the MIDI protocol which drives an electronic instrument through a sequence of simple "note-on" and "note-off" commands. The MIDI equivalents of traditional wind, bowed string, or brass instruments are still somewhat artificial sounding, (representing an interesting challenge for future generations of statisticians, among others). However, MIDI percussion instruments, piano for instance, and plucked strings

can be quite convincing. Luckily there is a significant amount of musical literature that features a solo instrument with piano accompaniment, providing a natural entry point for accompaniment systems. A welcome bonus of synthesizing the accompaniment with electronic instruments is the virtually unlimited technical capacity one inherits. In this way, compositions with nearly arbitrarily fast notes, arbitrarily complex rhythms, and superhuman complexity of interaction between live and synthetic players are now possible. We hope that this medium will successfully exploited by composers who remain engaged with traditional instruments, yet wish to combine new possibilities enabled by this technology. In fact, several such examples have already been written for our accompaniment system. One such example, the *Concerto for Accompaniment* by the talented young English Composer, Nick Collins, is a composition for oboe and piano six hands. While the piece is convincing on purely musical terms, the possibility of performance would be highly questionable without a computerized accompanist due to the nearly overwhelming technical demands for the "pianists." This work can be heard at http://fafner.math.umass.edu/chance as performed by the author with his accompaniment system.

An alternative method for producing the accompaniment audio is variable-rate resynthesis from an actual recording of the accompaniment, e. g. Music Minus One. Sampled audio provides a much richer and more nuanced sonic palette than does MIDI, and allows for a natural sounding full orchestral accompaniment of a soloist. Using techniques such as phase vocoding or synchronous overlap add (SOLA), one resynthesize a prerecorded sound file at variable rate without a corresponding change in pitch. The sequence of intermediate objectives generated by a scheduling mechanism, such as the one discussed above, function like a trail of breadcrumbs that guide the resynthesis throughout the performance. Two examples can be heard at http://fafner.math.umass.edu/chance which demonstrate this resynthesis technique. The first is an excerpt from Brahms' Violin Concerto beginning with the soloist's entrance, played by violinist Katherine Winterstein and a resynthesis of the Vienna Festival Orchestra. The second is roughly the first half of the 2nd movement of Rachmaninov's 2nd piano concerto with Greg Hayes on piano and accompaniment derived from a performance by the Stuttgart Symphony.. Both are beloved gems of the late Romantic concerto literature in which the inevitable interpretive liberties taken by the soloist demand considerable flexibility on the part of the orchestra. Both of the examples are taken from live recordings using the author's musical accompaniment system.

4 Will Musicians Embrace this Technology?

Given the largely distinct psychic spaces occupied by art and technology, it would be natural to expect musicians to show a certain amount of resistance to this technology. Some progress in this direction is evidenced by several commercially available accompaniment systems. However, the reliance on MIDI input, or lack of robustness in handling audio input, as well as mechanical-sounding musical output, will likely limit the range of musicians who are well-served by these systems.

In addition to the technical obstacles that still remain, we see two primary sources of resistance from musicians. The first is the tradition-bound nature of the classical musical world. While popular music is more eager to use new technologies, the tools of the classical musician have been comparatively static over the the last century. However, there are numerous examples in which technological progress has forever changed the musical landscape, such as fortepianos, valves on brass instruments, audio recording, and score-writing programs. We believe that, as with these other inventions, the potential impact of accompaniment systems will be significant enough to overcome the forces of tradition.

A more philosophical source of resistance lurks in the background as well, which might by expressed "Why dehumanize music with your cold and unfeeling computers?" The underlying suspicion here is that accompaniment systems attempt to substitute musicians with machines, thereby removing people from music-making. Our hope is that the opposite phenomenon will occur: accompaniment systems have the

potential to make music ultimately more accessible to the overwhelming majority of musicians who don't have a live accompanist or orchestra ready to play at a moment's notice. In this way we hope these systems will ultimately get *more* people interested and involved with music.

Our vision is that someday accompaniment systems will take their place as indispensable members of the musician's toolbox. In part, we see these systems as partners in teaching the technique of ensemble playing, much as the metronome helps with rhythm and the tuner with intonation. In part, we hope they will be embraced for the new kinds of music they can create. But, perhaps most importantly, by creating a laboratory in which the player can freely explore musical interpretation, the very soul of music-making, they bring immediacy and aesthetic satisfaction to a musician's practice. We can only hope that other musicians will find them as enjoyable and rewarding as we have.

References

[1] R. Dannenberg, (1984) "An On-Line Algorithm for Real-Time Accompaniment," *Proceedings of the ICMC, 1984* IRCAM Paris, France, 193–198, 1984.

[2] B. Vercoe, (1984) "The Synthetic Performer in the Context of Live Performance," *Proc. of the ICMC, 1984* 199–200, IRCAM Paris, France, 1984.

[3] Baird B., Blevins D., Zahler N., (1993) "Artificial Intelligence and Music: Implementing an Interactive Computer Performer," *Comp. Mus. Jour.*, vol. 17, no. 2, pp. 73–79.

[4] Raphael C. (1999), "Automatic Segmentation of Acoustic Musical Signals Using Hidden Markov Models," *IEEE Trans. on PAMI,* Vol. 21, No. 4, pp. 360–370.

[5] Orio, N., Dechelle, F., (2001) "Score Following Using Spectral Analysis and Hidden Markov Models", *Proc. of the ICMC, 2001* 151–154, 2001.

[6] Grubb L., Dannenberg R. (1998) "Enhanced Vocal Performance Tracking Using Multiple Information Sources," *Proc. of the ICMC, 1998* 37–44, 1998.

[7] Turetsky R., Ellis D. (2003) "Ground-Truth Transcriptions of Real Music from Force-Aligned MIDI syntheses," *Proc. Int. Symp. Music Info. Retrieval, 2003* 135–142, 2003.

[8] Soulez F., Rodet X., Schwarz D., (2003) "Improving Polyphonic and Poly-Instrumental Music to Score Alignment," *Proc. Int. Symp. Music Info. Retrieval, 2003* 143–150, 2003.

[9] Raphael C. (2001) "A Probabilistic Expert System for Automatic Musical Accompaniment," *J. of Comp. and Graph. Stats.* vol. 10, no. 3, 487–412, 2001.