Computer Coordination With Popular Music: A New Research Agenda[1]

Roger B. Dannenberg
roger.dannenberg@cs.cmu.edu
http://www.cs.cmu.edu/~rbd
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
phone: 412-268-3827
fax: 412-268-5576

Abstract. Computer accompaniment is now a well-established field of study, but nearly all work in this area assumes a Western classical music tradition in which timing is considered a flexible framework that performers manipulate for expression. This has important implications for the music, accompaniment, and performance style, and therefore the fundamental design of automatic accompaniment systems. Popular music has a very different set of assumptions and requirements for (human or computer) performers. In particular, the tempo is generally very stable, which would appear to simplify the synchronization problem, but synchronization must be quite precise and the music is not strictly notated, making coordination more difficult. The practical requirements for computer coordination with popular music create many problems that have not been addressed by research or practice. Some preliminary investigations in this area are reported, and future opportunities for research are discussed.

Introduction

Traditional computer accompaniment systems, dating back to early systems by Dannenberg (1989), Vercoe (1985), and Baird, et al. (1993) are based on music assumptions derived from Western classical music. In particular, expressive timing is important. Although this is thought by many to be “more advanced” music and a most demanding music for computer accompaniment, it is likely that the Western classical music has evolved to *enable* expressive timing (Temperley 2007). For example, classical music has relatively simple rhythms with a tendency to place note-onsets on beats, and this music is generally fully notated. These are properties that listeners and computer accompaniment systems can use to make sense out of musical signals.

In contrast, “popular music” (this term is used with some reservation and will be discussed further) tends to avoid expressive tempo variation, but in return for this simplification, popular music can introduce highly complex rhythmic patterns, elaborate improvisation, and a kind of expressive timing based on tight but carefully controlled synchronization between drums, bass, and other instruments and vocals. While current computer accompaniment systems can deal somewhat with popular music, they are really inadequate for serious music making. This creates an obvious research opportunity: characterize popular music performance, identify new methods for computer accompaniment, develop requirements, implement systems, evaluate new techniques, and explore artistic possibilities.

The following sections address these research ideas. Some interesting work has already started, including some study of actual music performance, but there is much more to be done. Many suggestions for future projects and research are included.

[1] Originally published as: Roger B. Dannenberg, “Computer Coordination With Popular Music: A New Research Agenda,” in *Proceedings of the Eleventh Biennial Arts and Technology Symposium at Connecticut College*, March 2008.

Let us return now to the term “popular music.” Using the word “popular” in a symposium devoted to pushing artistic boundaries should at least raise some eyebrows. Rather than defending the term, I will simply define what I mean by it. In this paper, “popular music” refers to music with these characteristics:

1. The music structure is “beat-based,” i.e. organized into beats and measures.
2. The tempo is mostly very steady, allowing timing of future notes to be predicted accurately from estimates of previous beat timings.
3. Some improvisation is expected, e.g. players might improvise chord voicings and rhythmic patterns, but there is enough predictability in the overall structure that it makes sense to synchronize additional computer music to a live performance.

These characteristics are often found in pop, rock, folk, musicals, and many other forms of music considered “popular.” However, not all pop, rock, jazz, etc. fit the definition. For example, free jazz many not meet any of the three criteria. Free jazz is not “popular music” in this context.

The Popular Music Augmentation Framework

Computers have many interesting applications in *live popular music performance*. In general, the goal is to add new sounds or to assist with mixing, digital audio effects, and sound reinforcement. Given the characteristics of “popular music” listed above, how can computers interact with live performers? A simple approach is to use a click track generated by the computer, but this is too restrictive and unpleasant for musicians. Essentially, what we want is a click track from musicians that the computer can follow. In addition to this framework of beats, we also need to cue the computer at some known point(s) to establish synchronization at a higher level of music structure.

Note how this plan uses characteristics of popular music: because the tempo is steady, it is straightforward to think about coordinating with beats, and this is possible even when the musical material is improvised (and might be difficult to follow directly).

In my current research, I am communicating beats to the computer simply by tapping on a foot pedal. (Dannenberg 2007) By giving a special pattern of taps, I can “cue” the computer to establish higher-level synchronization. This approach can be seen as an instance of a more general framework with the following components.

1. **Beat Detection and Prediction:** detect and track beat times and tempo by any means, including taps, beat detection from audio, drum sensors, etc. Tempo estimates are used to predict the times of future beats, which in turn enable the synchronization of computer-generated music and audio processing.
2. **Synchronization and Cueing:** cue the computer by some means to establish the score location, for example by pressing a button, music recognition, visual cues, vocal cues, or other gestures. For example, one might just cue the first note and count beats to maintain synchronization to the end of the piece. Alternatively, one might cue each musical entrance by the computer. (There might also be an intermediate level of synchronization where not only beats but measures or choruses are detected or cued so that the computer “knows” which beats are downbeats. This might simplify the cueing and synchronization problem.)
3. **The User Interface:** the musician coordinating with the computer needs feedback to confirm that the computer has locked onto the beat and knows the current score location. If the band or computer does something unexpected, the musician must use the interface to compensate or at least monitor the computer performance.
4. **Sound Generation and Processing:** Computers can augment performances in many ways. The computer can: (1) simply play additional parts through MIDI, (2) time-stretch pre-recorded audio to fit the current tempo, (3) adapt to other musicians to match tuning, vibrato, or articulations, (4) apply planned digital audio effects to various parts, (5) control a mixer based on knowledge of the

score location and intended balance, (6) create video projections, (7) play robotic instruments, etc.

Research Agenda

The brief outline of the framework hints at the rich research opportunities. In each of the four areas of the framework, there are many interesting, unsolved problems. Tempo detection through tapping or signal processing is not as simple as it may seem because beat-based music demands high precision, but it is not clear where to get an accurate and unambiguous indication of the "true" beat or whether such a thing really exists. Also, there is a conflict between responsiveness and stability when tracking beats and estimating tempo.

Synchronization and cueing can be done manually. For example, in a prototype system, I tap half notes on a foot pedal to establish the tempo, and I tap 4 quarter notes to cue an entrance. One could imagine taking cues by listening to other instruments, score following, or even visual cues. The biggest challenge is to find a technique that is generally applicable and highly robust in real performance situations.

The user interface must be used in real time, possibly by a musician who is performing on an instrument at the same time. One form to consider is the digital display of common practice music, augmented with an indication of the computer's state, with interaction through a touch screen or other sensors. Many other forms are possible. In addition to the user interface, one must consider how material is *prepared* for performance. E.g. for a digital music display to be practical, there must be support tools to allow musicians to capture printed music and relate the music image to the musical structure and material to be synchronized by the computer.

The area of sound generation and processing is perhaps the most interesting one of all. Given a structure in which the computer can follow popular music, what can the computer do to augment the performance? The original motivation for this work was the idea that the computer might perform additional parts, especially difficult ones such as finishing on a very high note. After further consideration, there are innumerable possibilities. A partial list includes digital audio effects, mixing for better balance, intonation adjustments, lighting control, control of robotics and animation, virtual orchestras, "virtual subs" to facilitate rehearsals when a band member is absent, and "music minus one" for practice at home.

Thus, there are many interesting ideas to pursue in at least four distinct areas within the Popular Music Augmentation framework. The next section will describe some preliminary investigations into the first area, of beat detection and prediction. I hope that other researchers will begin to work in this area and perhaps coordinate to explore the field as widely and deeply as possible.

Beat Detection and Prediction in Popular Music

One might expect that generating a "click track" for popular music is simple. For example, one could simply tap beats on a foot pedal. To a first approximation, this is true. It is not hard to get beats from a foot pedal, and with a little smoothing, one can estimate the tempo and predict future beat times. The difficulty is getting *accurate* times in a *real* performance setting, especially from a musician who might be concerned with other difficult tasks (including playing an instrument).

Timing in popular music is critical, and large errors of, say, 50ms or more, are not likely to be acceptable. Figure 1 shows a plot of consecutive beat durations measured while the author simultaneously played trumpet in a big band and tapped on a foot pedal. The task did not seem difficult, but the variation in beat times is considerable. The data and listening suggest that the tempo is actually quite steady and that the variation in the figure is due mostly to imprecise tap timing. This is a combination of inherent human limitations, ergonomic issues (the measured contact time may not correspond to the perceptual tap time), and both cognitive and motor interference with the trumpet playing task. Regardless of the source, this is pretty noisy data for fast, reliable tempo estimation.

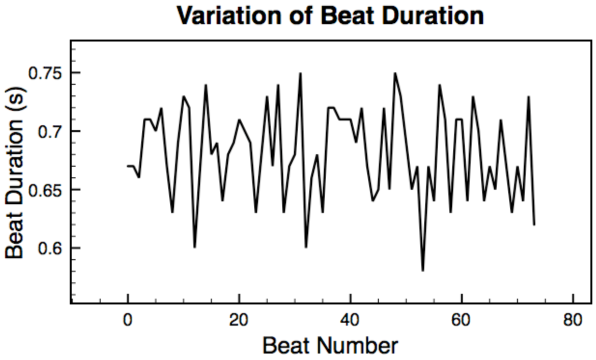

Figure 1. Time between successive taps, showing large variation in measured data. The mean is 0.681s; the standard deviation is 0.039s.

Smoothing is one way to deal with noisy data. In fact, it seems that the long tap intervals are quickly compensated by short ones (otherwise the taps would drift out of synchronization with the band, but this does not happen). Therefore, just the average tap time within a window is one way to get a better tempo and beat location estimate. Figure 2 shows the average of the same data within a 20-beat smoothing window.

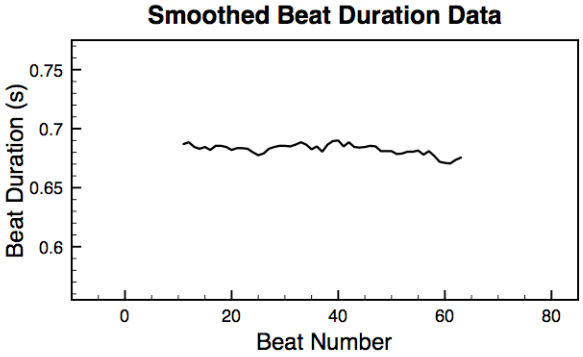

Figure 2. Data from Figure 1, averaged over a 20-point smoothing window.

This looks promising: the variation now is only from 0.67s to 0.69s, a range of only 20ms. However, smoothing will make the system less responsive if the tempo changes. In short, the smoothed prediction will be based on the old tempo rather than the new tempo. How much does tempo change in popular music performance?

To answer this question, I tapped a foot pedal during rehearsals with a big band and a jazz octet. In all cases, the tempo was nominally constant (no notated tempo changes). In practice tempos change because band members or the band leader feels the initial tempo is not correct. Musicians tend to rush during exciting moments and drag for example when music is loud or very technical. Figure 3 illustrates an excerpt from over 3 hours of recorded taps on various songs.

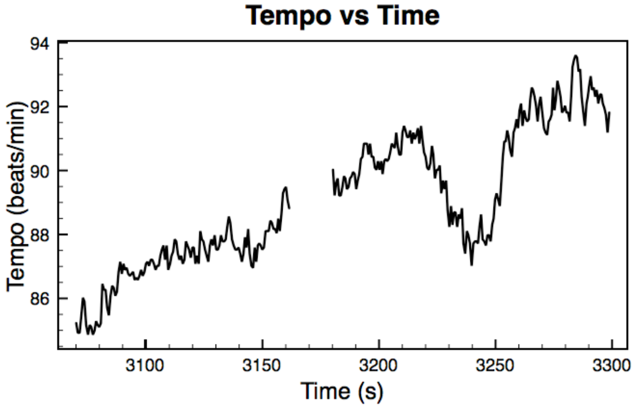

Figure 3. Tempo estimates from smoothed foot-tap data as a function of time. Duration is 229s for 316 taps on half notes (the meter is cut time, i.e. 2/2), or 168 measures. Data in the middle is missing due to missing tap data.

This example was chosen for its several interesting features. First, there is almost a constant acceleration, from about 85 to 92 beats per minute. Over the course of almost 4 minutes, this is a very slow rate of change, but it illustrates the problem of assuming a constant tempo. Second, there is a gap in the middle of the data. This is due to a problem with a few taps (affecting 25 or so smoothing windows). Any tap-based system needs to tolerate a foot landing in the wrong spot, accidental double taps, etc. Finally, there is a dip in the tempo around 3240s. Listening reveals that there were changes in the music at this point and that the band did indeed slow down a bit. At around 3260s, the rhythm section pulled the tempo back up and, collectively, the band resumed its gradual accelerando.

The slow-down section is interesting because it gives a sense of how rapidly a tempo following system must respond to tempo change. We can fit a curve to just the data around 3240s and measure the rate of tempo change. Figure 4 shows this.

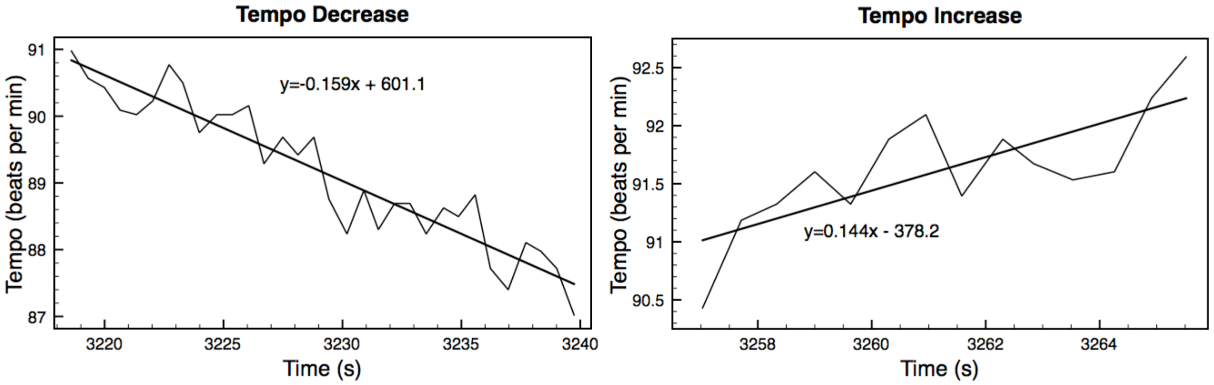

Figure 4. DetailS from Figure 3 with least-squares linear fit to estimate rate of tempo decrease and increase.

From the equations in Figure 4, we can see that the tempo changes by about 0.15 bpm per s. (A strange unit indeed.) This is interesting because it gives a sense of how rapidly a tempo following system must respond to tempo change. Suppose, for example, that we estimate the tempo on the basis of a linear fit to 20 previous points (many other schemes are possible). How much error will be introduced by a tempo change? A simple discrete time simulation shows that at the 21st beat, the tempo will be 92.1 for a beat duration of 0.652s. The previous 20 notes spanned 13.20s for an average duration of 0.660s. Thus the predicted beat times will differ from the “true” beat by about 8ms. This is quite a low number, with the just noticeable difference for time intervals like this in laboratory conditions being about 10ms.

What have we learned from all this? First, direct data from pedal taps is likely to be too imprecise for musical applications. Our sample population (the author) was only one, but the data seems to be representative of what one might expect from others. Estimating beat locations and tempo from noisy input data like this calls for smoothing, but smoothing runs the risk of making the system unresponsive. By measuring bands in realistic performance situations, we found that the rate of tempo change is actually quite small. If this is typical, then even smoothed data will not deviate too far from the “truth” when the tempo is changing. This is promising, but not the whole story.

A question that is not answered in this study is: How much jitter in tempo (or beat duration) is tolerable in a performance? Perhaps even longer smoothing windows or more sophisticated tempo models will be necessary, making the system less adaptable to tempo change. Also, there may be performance situations where tempo change is much greater than what I observed. How can we make the system robust enough for more extreme cases?

Conclusions

The Popular Music Augmentation Framework proposes a new kind of interactive music system based on current practice in popular music performance. This practice has been ignored for the most part by computer music research, and consequently, there are many interesting and wide-open topics for study. I have shown how the simple question of how to build an adaptive tempo follower has led to interesting studies and measurement of real music practice. The results are encouraging that fairly simple interfaces can be used to generate an adaptive click track for computer synchronization, but testing these conjectures will require much more work. The real test of these ideas will be in the context of a complete and working system. Such a system will have to solve the problems of cueing the computer, interfacing with the user/musician, and generating musical output or control. All of these facets of the framework are open for exploration and creative innovation.

References

Baird, B. Blevins, D., and Zahler, N. Artificial Intelligence and Music: Implementing an Interactive Computer Performer. *Computer Music Journal 17*, 2, 1993, 73–79.

Dannenberg, R. B. Real-Time Scheduling and Computer Accompaniment. In Mathews, M. and Pierce, J. eds. *Current Research in Computer Music*, MIT Press, Cambridge, 1989, 225-261.

Dannenberg, R. B. New Interfaces for Popular Music Performance. In Proceedings of New Interfaces for Musical Expression 2007, New York University, New York, 2007, pp. 130-135.

Temperley, D. *Music and Probability*. MIT Press, Cambridge, 2007.

Vercoe, B. The Synthetic Performer in the Context of Live Performance. In Proceedings of the International Computer Music Conference 1984, (Paris). International Computer Music Association, San Francisco, 1985, 199–200.