# New Interfaces for Popular Music Performance

Roger B. Dannenberg
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
+1 412 268 3827

rbd@cs.cmu.edu

## ABSTRACT

Augmenting performances of live popular music with computer systems poses many new challenges. Here, "popular music" is taken to mean music with a mostly steady tempo, some improvisational elements, and largely predetermined melodies, harmonies, and other parts. The overall problem is studied by developing a framework consisting of constraints and subproblems that any solution should address. These problems include beat acquisition, beat phase, score location, sound synthesis, data preparation, and adaptation. A prototype system is described that offers a set of solutions to the problems posed by the framework, and future work is suggested.

## Keywords

Accompaniment, music synchronization, conducting, beat tracking, virtual orchestra, synthetic performer, intelligent performer.

## 1. INTRODUCTION

Computer music systems developed by the academic and arts communities have largely ignored popular music performance as an interesting area for research. Few if any systems are really appropriate for most popular music settings. Of course, one could say the commercial sector has had a significant impact on popular music through drum machines, sequencers, and loop-based interfaces. However, it seems that popular music has adapted to new technology rather than the other way around. For example, the precision of drum machines seemed stiff, mechanical, and monotonous to many musicians, but that became the trance-like foundation of club dance music and other forms. Similarly, the inability of sequencers and other beat-based software to "listen" to human musicians has led to performances with click tracks or simply a fixed drum track that live musicians must follow.

Without taking a position for or against these recent developments, the current work aims to explore interfaces and techniques that allow more flexible interaction with live musicians in popular music settings. A new framework called *PMA,* for *Popular Music Augmentation*, defines a range of music performance styles where PMA applies, establishes a set of constraints within which PMA operates, offers a set of specific problems to be solved, and proposes a set of solutions and future research directions. Although this work includes some specific

hardware, software, and evaluation, the main goal is to introduce the PMA framework as a new paradigm or category of interactive music systems.

Categories and labels for music are risky, but the term "popular music" was chosen as the best way to describe music where this work applies and to make the point that this work is *not* appropriate for *all* music. In this context, "popular music" is intended to mean music that is organized around a steady beat structure, has at least many parts that are notated, is performed live, and may have repetitions or omissions of sections determined during the performance. Most rock, folk, and musical theater, and some jazz falls into this category.

The next section outlines some ways computers can augment popular music and describes some constraints and considerations. Section 3 presents a set of specific problems that must be solved. Section 4 presents some solutions and work in progress as well as some evaluation. Section 5 presents conclusions.

## 2. INTERACTION IN POPULAR MUSIC

Computer Accompaniment [7] is a good example of augmenting music performances. Computer Accompaniment assumes a score the computer can follow as a human performs expressively. The computer synchronizes an "accompaniment," which may be any musical material, to the human player.

Popular music does not fit the framework of Computer Accompaniment in several respects. First, popular music is partially improvised. In particular, keyboard and guitar players play chords but may voice chords differently or change the rhythms from one performance to the next. Similarly, drummers rarely play from detailed notation. Second, popular music allows for large structural changes: a band might vamp until the lead vocalist enters with the melody, a chorus might be repeated based on the audience response (or a mistake!), and at various places there might be improvised solos.

What opportunities exist for "augmentation?" This work is mainly concerned with adding musical lines, creating artificial, synthetic or "virtual" performers that add new parts to the live music. An example that has driven most of this work to date is the horn player in a rock band. ("Horn" refers to any wind instrument.) In many cases, horns play as a section and do not improvise, so there is a good opportunity to have a computer play one or more horn parts. Horns have distinctive radiation patterns in acoustic settings that make replacement with electronics especially difficult, but with rock music, even live horn players are amplified. It should be possible to create convincing computer-generated horn parts that blend in with other horns and the band. Computers can also play a more central role, playing drums, guitar, keyboard, or even vocals. Computers might also provide parts that are not normally available in live performances, e.g. a string orchestra augmenting

a jazz big band, or electronics and studio effects augmenting and/or manipulating the sound of a live rock band.

One might complain that, for example, augmenting a rock band with horns or strings is an uninteresting throwback, or even that live drumming has given way to DJs, tape, and sequencers. While these are interesting points, consider that many popular music styles are flourishing simultaneously today, and techniques inspired by more traditional forms will enable new musical directions in the future. Solving some fundamental problems of human-computer interaction can have many benefits.

## 3. RELATED WORK

As mentioned, computer accompaniment systems come close to the task of augmenting popular music, and there have been attempts to use computer accompaniment for interactive Karaoke. (See http://www.media.mit.edu/sponsors/sc-taito.html) Following a vocalist is appropriate for some musical situations, but it is often the case that the vocalist works around a rhythmic foundation set by the band, and there has been little study of how the musicians and systems can decide who is leading and who is following.

Rock bands such as the Who played with tape and sequencers at least as early as the 70's, but not without problems. In spite of some successes, the Who's Quadraphenia tour was plagued by continuous difficulty with many backing tapes, eventually causing the group to abandon the opera in live performance. (C.f. http://www.quadrophenia.net/1973tour/uk.html) The present work introduces new techniques that support and extend the practice of augmenting live performances with "taped" music, striving for greater control, reliability, and flexibility.

Cirque du Soleil makes creative use of MIDI sequencers to augment their live orchestra. A click track is used to establish the beat, but the score has many vamps (short sections that can be repeated indefinitely) so the musicians can adapt to the on-stage circus performance. The music director triggers MIDI sequences by playing keys on a keyboard exactly in rhythm, and sequencer controls can be used to alter the tempo if desired. The live musicians and singers are cued verbally over headsets, and on-stage singers wear wireless in-ear monitors and wireless microphones. This blend of synthesizers and live players works very well, but it is adapted to a particular form of music and relies upon click tracks and verbal communication to coordinate the orchestra members.

Conducting systems [5], [10] generally use tapping or a gestural sensor of some kind to obtain the beat from a conductor, who might also give cues to control score location. Conducting systems have been used to synchronize MIDI sequences to ensembles. More recently, this technique has been used for opera and musical productions, often with great controversy. [12]

Another music augmentation system, developed by Music Arts Technologies, is widely used in Broadway musicals and relies on keyboard players to play essentially all the notes, but maps the performed notes to multiple channels of synthesizers. The typical application is to make a small touring ensemble sound more like a full-sized orchestra. In this system, the mappings change frequently, and the musicians have the responsibility of keeping the current mapping synchronized to section numbers notated in their score. A small box with some buttons and a numerical display is used for this purpose.

Beat and tempo detection are well-studied topics [2][9], but seem to be used only rarely in live performance, an indication that sufficiently accurate and robust systems do not yet exist.

Tempo and beat synchronization are supported in some software, especially Ableton Live (www.ableton.com), but the interface does not seem to be designed to synchronize to live musicians.

## 4. A PROBLEM FRAMEWORK

Given the goal of augmenting live popular music performances, there are many different approaches. To focus the discussion, however, we will narrow the solution space by developing a framework of assumptions and approaches. The hope is that this framework, called *PMA* for *Popular Music Augmentation* will be general enough to let us consider and develop many promising technologies, but narrow enough to suggest productive paths for research and development.

The first assumption is that we want to obtain a high degree of autonomy. A control interface that requires the full attention of a human is either a form of musical instrument or a conducting system. These systems are interesting in their own right, but are excluded deliberately from the PMA framework. Instead, we want something that can be operated by a musician (band member) who is already occupied by ordinary music performance tasks.

The second assumption in our framework regards tempo. We assume three rough categories for music: (1) fixed tempo (determined by click track, drum machine, etc.), (2) steady tempo (characteristic of live popular music), and (3) expressive tempo (romantic period music, for example). Our framework focuses on *steady tempo music*. This is an interesting middle ground where systems must adapt to live players, but where tempo is steady enough that beat-based synchronization is possible.

Third, the perception of this tempo may be difficult to automate. Popular music often uses syncopation and deceptive rhythms that place musical events *around* the beat as much as on it, so automated beat tracking should not be the primary source for beat information.

Fourth, because the structure of the live performance is not completely determined, there must be ways to communicate changes to the computer. For example, if a section is repeated, the computer must "know" this in order to continue playing with the band. Detecting the unfolding structure can use a combination of automated machine listening and explicit cues from live players or conductors.

Given this framework, we obtain a set of problems: acquiring the beat, acquiring the beat phase, maintaining score location, sound synthesis, data preparation, and on-line adaptation. These are described in the following paragraphs.

### 4.1 Acquiring the Beat

It is assumed that the beat is the primary means for synchronizing to live music. The beat can be indicated to the computer by tapping, using any number of sensors for heel or toe, fingers, head, etc. For some music, it might make sense to track a drummer, another instrument, or combination of instruments, but the third assumption says that there must be some way to correct or override automatic tracking systems. Sensors for beat tapping must be easy to operate while playing an instrument, and the beat detection must be reliable.

## 4.2 Acquiring the Beat Phase

Popular music is organized into measures and phrases above the beat level. Visual or vocal cues between musicians are often not accurate to the beat but instead refer to the next measure, next phrase, or next section. If the computer system has a similar notion of measures and higher-level structure, it can eliminate the need for operators to give cues that are accurate to a fraction of a beat. For example, if the system keeps track of measures, then a cue to enter at a particular phrase can be given up to one measure in advance. Because this timing need not be precise, interesting cuing mechanisms are possible such as raising a horn to a player's lips, eye tracking, breathing, and other gestures that are natural for the player but perhaps not precisely timed. This is called "phase" because it refers to progress within a (mainly) cyclical structure such as a measure, a repeating chord progression, or 8-bar phrase.

Having a beat-phase representation means that there must be a way to communicate phase to the computer and there must be a way for users to monitor that the computer's representation of phase is correct. Communicating phase to the computer can be accomplished by introducing asymmetry into beat tapping. For example, by tapping only beats 1, 3, and 4, it is relatively easy to acquire not only the beats, but also the position within a 4/4 measure. Another form of asymmetry uses a different form of tap to indicate measure downbeats, section beginnings, etc. This second form, which might be a keypad for example, is only necessary to correct the system and therefore might be used only occasionally.

Monitoring the beat and phase can be accomplished through visual, aural, or tactile means. Visual displays have the potential for showing detailed information (current measure number, beats and phase), but may be difficult to monitor while reading music. If music is integrated into a computer display [1], there is the potential to display beat information right on the music notation. Finding effective ways to do this is an interesting design problem worth further study. Aural displays could be distracting in a live performance, but listening to the computer's notion of beats mixed with live music over headphones offers a very precise way to determine if the computer is "in time" or not. Tactile displays are an interesting possibility because they might avoid overloading the musician's visual and aural channels. A comparative study is suggested by the various possibilities.

## 4.3 Maintaining Score Location

Ultimately, acquiring beats and beat phase are just enabling steps to achieve complete synchronization to the live music. Simply put, the important thing is to play the right thing at the right time. If the band plays a predetermined score from beginning to end, then the problem is only to synchronize on the first beat and count beats until the end. This is why beat acquisition is important. However, our fourth assumption is that the live performance is not completely determined. Thus, there will be times where the computer must be updated with a new location, and there are many possible ways to do this.

One idea is to simply cue every entrance. In this approach, the computer indicates the next phrase to be played, perhaps by displaying a number or letter also indicated above the music notation. If the music is displayed digitally, various forms of shading and highlighting might be used to indicate positions in the score. Some cuing mechanism is required. This might be an alternate form of tap, or some gestural input, keeping in mind that

the musician is already dealing with the task of playing an instrument. (The role of beat phase should now be even more clear: The musician can cue computer entrances at least several beats ahead, allowing the musician to focus on making the same entrance with his or her own instrument.)

The computer can alternatively count beats and make all entrances automatically. Here, feedback to the musician is critical to avoid wrong entrances. Whether each entrance is cued or not, it is important that musicians have quick and easy ways to update the score location to handle changes in live performance and also to facilitate rehearsals, where the band will normally restart from various score locations. If score locations and entrances are indicated by numbers or letters, updates might be entered on a keypad or adjusted with a touch sensor or foot switches. If the score is displayed electronically, the musician can simply point to the current location on a touch-sensitive display.

If a wrong cue is given or if a cue is misinterpreted, there is the risk that the computer will play at the wrong time. The PMA system should indicate that an entrance is imminent, and there should be an easy-to-operate override to mute the output in case of a mistake. Recovery may be a simple matter of correctly cuing the system on the next entrance. If the computer should be playing an extended passage, automatic score following techniques might allow the system to find the correct score location automatically and recover more quickly. Combining score-following location information with accurate beat and beat-phase information is a new area for exploration.

## 4.4 Sound Synthesis

Having acquired instructions to play a particular part at a particular time and at a particular tempo, the computer system needs to generate that part. One method is to use MIDI synthesizers. This is very simple, but it will not be very satisfying for horn lines and many other instruments. Because parts are known in advance, a good way to generate parts is to pre-record them and use time stretching techniques to adjust the tempo.

Time-stretching can lead to annoying artifacts, and the best algorithm will depend upon the source. In this framework, the time stretching requirements are slightly different from those of typical applications. Here, the sound is completely determined (pre-recorded) and thus subject to off-line analysis. On the other hand, the amount of time stretching is continuously variable because the tempo may change slightly in real time. Finally, the results must be delivered in real time.

In many cases, the band will want to augment its sound using instruments that the band members do not play. This presents a new synthesis problem that has received little attention to date. The problem is, given a full score, synthesize a convincing musical performance. The synthesizer can look ahead in the score and also use phrase markings and other information not available to, say, MIDI synthesizers. Work such as Synful (www.synful.com) and CSIS [6] synthesizers indicate that vast improvements in realism can be obtained under these conditions.

## 4.5 Data Preparation

To prepare for augmented performances, the user must record or create new parts that the computer will play, organize these parts within a score, and organize scores into libraries or set lists. In addition to score preparation, there must be rehearsal markings or section labels to allow musicians to coordinate printed music with

the computer performance and mechanisms for maintaining score location, cuing entrances, etc.

The preparation of all this data might be an exhausting task. It seems likely that there will be a need for a software package to coordinate music notation, music recordings, labels, tempo markings, cues, and other annotations. Probably this software should manage both data preparation and music performance, supporting a smooth transition between the two modes during rehearsals. An opportunity exists to integrate PMA functions with those of an electronic music display [1], creating an innovative new product category for musicians.

## 4.6 On-Line Adaptation

Already, we have seen a variety of problems motivated by this music system framework. These problems must all be addressed just to accomplish the basic level of functionality implied by the PMA framework. Once this is accomplished, there will be many ways to enhance and improve the computer side of the music performance. Dynamics is important. In a rock band, a sound engineer rather than the band itself often controls sound levels, so automation may not be a big consideration. But, for example, if the computer generates horn parts to augment existing horns, matching dynamics to the live players may be critical for a good section sound. It might make sense to send a submix of the live players through the computer system where synthesized parts can be added after sensing the signal level of the live horns. Here, the computer might also process the live sound using compression, equalization, reverberation, and other effects which can depend on the song and score location since these are known.

The computer can adapt in other ways. The explicit indications of score location can be used to train the system to listen to and follow the fully polyphonic audio produced by the band, possibly leading to greater redundancy and reliability in performances. Even explicitly tapped beats might be consistently ahead or behind the beat, and comparing these taps to acoustic signals such as a snare drum microphone might lead to better synchronization. Notes may not be played exactly on the beat anyway, so in the case of a horn section, the computer could compare the synthesized output of horn parts with audio signals from live horns. If there is a systematic anticipation or delay, the system could automatically correct for this to make the section "tighter."

Intonation is another area where interesting algorithms might be developed. The computer should have no trouble playing in equal temperament, but particularly with horns and strings, it might be better to adapt to live players. Examples include playing (nearly) exact unisons and octaves, beat-free major thirds, raised leading tones, and Pythagorean tuning.

More radical manipulation of the live sound with effects processing and hyper-instrument-like [11] augmentation might also take advantage of score location, tempo, and beat phase.

## 5. Current Work

A prototype system has been constructed based on the framework presented above. This system makes many specific design decisions and therefore represents just one of many possible approaches that are suggested by this framework. The prototype is designed for use by the author, a trumpet player, allowing him to augment his own playing with one or more synthesized trumpets. A PMA system that can take over any part on demand would

enhance the overall sound, facilitate rehearsals, and make playing more comfortable and enjoyable.

Tapping seems to be the simplest approach to acquire the beat, but making this easy and reliable raises many design questions. Personal experience with foot switches in live performance provides plenty of evidence that: (1) switches are hard to find on the floor when you are reading music, (2) switches are easy to press by accident as well as easy to press too softly, and (3) pedals requiring up-and-down or back-and-forth action while the operator stands mostly on the other foot are awkward. By playing trumpet and thinking about what kinds of tapping are natural, I arrived at a configuration where I tap by applying pressure with my heel on a flat, thin panel. This allows me to keep weight on the ball of my foot and maintain my balance, it does not require much actual motion, and the joint angles and muscle tensions all remain in a comfortable range. (Guitar players accustomed to pedals are surprised by this configuration, suggesting that different players and instruments demand different interfaces.)

## 5.1 The Pressure-Sensing Pedal

A custom-built pressure-sensitive foot pedal was constructed to enable this form of input. The pedal (see Figure 1) uses a sandwich construction. Two outer panels of thin plywood make the pedal very strong yet slightly flexible. At the center is a force-sensitive resistor. To transfer pressure evenly from the plywood to the FSR and to protect the FSR from being crushed, the FSR is sandwiched between two layers of very thin foam packing material.
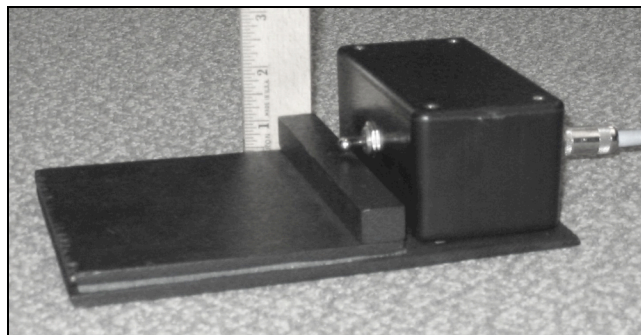


**Figure 1. Pressure-sensitive foot pedal. FSR is sandwiched between plywood and foam at left. Electronics and battery power in box at right. Output signal is through a ¼-inch phone jack in back.**

The interface from the FSR to the computer system is perhaps novel but quite simple. The FSR participates in an RC circuit that determines the frequency of the well-known 555 timer chip. The output of this chip goes directly to a computer audio input channel. After digitization, the period of the signal is detected reliably simply by looking for positive zero crossings. The frequency range is roughly 0.4 to 9kHz, corresponding to periods of 110 to 5 samples at a sampling rate of 44100 Hz.

## 5.2 Beat Detection

In normal use, the pedal is used to tap half notes. A tap corresponds to a change in 555 output frequency. To avoid glitches, the minimum period is computed for every block of 256 samples. When the minimum period is greater than 20 samples in one block of 256 samples, and less than 20 samples in the next block, a tap is reported.

Further processing estimates tempo and beat detection. At every tap, software looks to see if the two previous taps and the new one are equally spaced (within some threshold). If so, we say that the last tap indicates a beat at a tempo indicated by the average spacing of the three taps.

The computer maintains a model of current tempo and beat phase so that it can fill in when beats are missing and so that tempo can be smoothed over time to minimize the effect of early or late beats due to human error. When three taps indicate a new beat and tempo, the computer's model is modified by adjusting the tempo in the direction of the new tempo indication and adjusting the beat phase in the direction of the latest tap. Limits can be placed on the amount of adjustment to prevent accidental taps from having a large influence on the beat model.

This approach is simple, it acquires the tempo quickly, and it maintains tempo reliably. Small tempo changes or phase drift are easily and quickly corrected. Probably, acquisition could be even faster and the beat detection could be more resilient to errors with other schemes. This is an interesting problem for research.

## 5.3  Beat Phase and Score Location

In the current prototype, taps are on half notes. To cue an entrance (and also indicate the phase within the measure), the musician makes 4 taps on quarter notes. Since the musician is tapping on beats 1 and 3 prior to the 4-beat pattern, the cue pattern corresponds exactly to the familiar count-off: "1 (pause) 2 (pause) 1 2 3 4 ". Note that the musician always cues the beginning of the measure where playing begins, even if nothing is played on the downbeat. The actual entrance time within the measure is determined by the score. Also, this scheme can be adapted easily to 3/4 time by tapping on beat 1 to establish the tempo and tapping on 3 quarters to cue an entrance.

Since the user taps on consecutive quarter notes to cue an entrance, these quarters could be interpreted as half notes at twice the tempo. Confusion is avoided by first establishing the tempo with half notes. Then, if a tap arrives half-way between half-note taps, it is assumed to be a quarter note tap and is not considered in the search for taps that establish the tempo.

Additional interaction will be necessary to indicate the score location if sections are not played in the expected sequence. Figure 2 illustrates a planned interface for the laptop computer that will sense pedal input and generate audio output. This is considered as a "backup" or "system override" interface that demands more user attention, but will be rarely used.
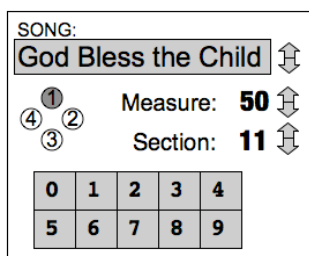


**Figure 2. Example graphical user interface to select songs and sections of songs and to give feedback and beat, measure, and section confirmation to the user.**

## 5.4  Sound Synthesis

Trumpet lines are synthesized by processing pre-recorded lines. The first step is to label the trumpet performance with beat locations. This can be done manually, by score following [8], or by simply recording against a click track so that the beat locations are known in advance. The second step is to analyze the performance with a pitch estimator. Currently, the YIN algorithm [3] is used because it exists as a primitive in the Nyquist signal processing language [4] and because YIN outputs a confidence level as well as a pitch estimate. Third, regions of steady pitch with high confidence are identified. These will become the regions that are stretched, while the other regions consisting of attacks and slurs will be untouched. To stretch the sound, we allow any period in a stretchable region to be duplicated one time, effectively doubling the duration. To shorten the sound, we allow up to every other period to be dropped, effectively halving the duration.

By dropping or duplicating whole periods, we eliminate the problems of phase cancellation that occur when overlap-adds are used either in the time domain or when reconstructing from spectral frames. Since a high proportion of the sound is expected to consist of stable pitch regions and the amount of time stretching is likely to be small, this approach fits within the proposed framework.

For real-time performance, audio must be computed at least a few milliseconds in advance, but since tempo is known to be nearly constant, timing can be predicted accurately to allow computation to run ahead of real time. To date, this time-stretching method has been prototyped and tested off-line. Future work will integrate sound generation with other components into a real-time system.

## 6.  EVALUATION

As a preliminary but fairly objective test of the beat acquisition system, I tapped along to a 2-minute "click track" consisting of a click every quarter note. The click track changes tempo smoothly from 80 beats per minute to 160 beats per minute. Software records the difference (error) between the actual and acquired beats, starting after a 10-second introduction. The standard deviation of the time differences was 130 ms. However, when parameters were altered to make the tempo and beat phase more stable, and the click track was a constant 120 beats per minute, the standard deviation dropped to a respectable 28 ms. (A general rule of thumb is that the just-noticeable difference for tapped rhythms is around 10 ms.) Based on this preliminary data, synchronization works well with a steady tempo, but more work is needed to obtain tight synchronization when the tempo is allowed to vary.

These tests were not conducted while playing trumpet at the same time. It is predicted that trumpet playing or any other task will degrade the performance somewhat, but that practice with tapping will improve performance. One should also expect individual differences, so the measurements only serve to indicate the general performance of the foot-pedal approach.

## 7.  SUMMARY AND CONCLUSIONS

The goal of this paper is to introduce a new approach to making music with computers. Although there is considerable overlap with existing technologies and music practice, the overall problem framework differs significantly from others known to the author. The task addressed here, called *PMA* or Popular Music Augmentation, is to augment live popular music performances with additional musical parts without using a click track,

sequencer, or tape. The added parts should be created with the highest musical quality, for example applying high-quality time stretching to studio-quality recordings.

While this task is simply stated, there are many problems and many more potential solutions and approaches. Rather than thinking of this as a specific technology or system, it is helpful to regard the general task as providing a *framework* within which various designs and solutions are suggested and explored. Thus, a set of problems is presented, including beat acquisition, obtaining beat phase, score position, synthesis, data preparation, and adaptation. In general, multiple solutions to these sub-problems exist and may be combined in many ways. The PMA framework is a rich source of future research problems.

It should be noted that cuing audio into a live performance is a general technique with many applications. On the other hand, working within the confines of the PMA framework suggests many techniques that remain to be implemented. The central problem is largely a human-computer-interaction problem: How can a musician communicate beat and score location information to a computer and monitor that the computer's real-time information is correct, all without overloading the musician who is simultaneously playing a traditional instrument? This problem is closely aligned with the concerns of the NIME community.

As a first attempt to solve these problems and build a complete and operational system, the author has created hardware and software as a personal system for augmenting trumpet performances in a rock band. This system uses a pressure-sensitive pedal to mark half notes, and four quarters serve as a cue. Pre-recorded trumpet parts, labeled with beat locations, are time-stretched to align to the tapped tempo using a technique that doubles or drops whole, exact periods to minimize any signal-processing artifacts. Experiments show that this prototype in fact performs reliably with minimal impact on the musician's ability to play trumpet while controlling and cuing the system. Testing in a real concert situation will require the completion of a suitable interface for song selection and to indicate deviations from the score (such as inserted measures or extra repetitions.)

In conclusion, there are many variations on PMA systems within the general PMA framework. Some systems may choose to place more load on the musician in return for more control over the computer systems. Some systems may focus on MIDI sequences for orchestration flexibility or even improvisation while others will strive for the highest quality simulation of acoustic instruments. Some will focus on making tapping easier and more reliable, while others will tie into MIDI drums or keyboards to enable the extraction of beat and score location through machine listening. These suggestions give some idea of the richness of this framework for thinking about a particular range of musical interactions. Experimentation with these and other variations is quite a large research agenda.

It is interesting that after decades of interactive computer music system development, we as a community have not developed a system that can autonomously perform with live musicians within the norms of popular music performance, including such forms as rock, jazz, or folk. The goal of this paper is to undertake an analysis of the popular music augmentation task, develop a framework within which solutions can explored, and to illustrate at least one set of viable solutions through a set of

implementations. I hope that other researchers will see the value of this framework and join me in further exploration.

## 8. ACKNOWLEDGMENTS

## 9. REFERENCES
[1] Bell, T., Blizzard, D., Green, R., and Bainbridge, D. Design of a digital music stand. In *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR06)* (London, Sep. 11-15, 2005), Queen Mary, University of London, 430-433.

[2] Brossier, P., Bello, J. P., and Plumbley, M. D. Real-time temporal segmentation of note objects in music signals. In *Proceedings ICMC 2004*, (Miami, Florida, Nov. 1-6, 2004), International Computer Music Association, 458-461.

[3] de Cheveigne, A., and Kawahara, H. YIN, a fundamental frequency estimator for speech and music. *Journal of the Acoustical Society of America* 111, 4 (Apr. 2002), 1917-1930.

[4] Dannenberg, Machine tongues XIX: Nyquist, a language for composition and sound synthesis. *Computer Music Journal*, 21, 3 (Fall 1997), 50-60.

[5] Dannenberg, R., and Bookstein, K. Practical Aspects of a midi conducting program. In *Proceedings of the 1991 International Computer Music Conference* (October 1991). International Computer Music Association, San Francisco, CA, 1991, 537-540.

[6] Dannenberg, R., and Derenyi, I. Combining instrument and performance models for high-quality music synthesis. *Journal of New Music Research*, 27, 3 (Sep. 1998), 211-238.

[7] Dannenberg, R., and Raphael, C. Music score alignment and computer accompaniment. *Communications of the ACM, 49, 8* (Aug. 2006), 39-43.

[8] Hu, N., and Dannenberg, R. Bootstrap learning for accurate onset detection. *Machine Learning*, 65, 2-3 (Dec. 2006), 457-471.

[9] F. Gouyon, F., Klapuri, A., Dixon, S., Alonso, M., Tzanetakis, G., Uhle, C., and Cano, P. An Experimental Comparison of Audio Tempo Induction Algorithms. *IEEE Transactions on Audio, Speech and Language Processing*, 14, 5 (Sep. 2006), 1832-1844.

[10] Lee, E., and Borchers, J. The role of time in engineering computer music systems. In *Proceedings of the 2005 International Conference on New Interfaces for Musical Expression* (NIME05), Vancouver, BC, Canada, 2005, 204-207.

[11] Machover, T. Dreaming a New Music, *Chamber Music, 23, 5* (Oct. 2006)*, 46 - 54.

[12] Shulgold, M. 'Virtual orchestra' breeds real rage from musicians. In *Rocky Mountain News.* Dec. 18, 2004. Online at: http://www.dmamusic.org/news/000198.php.