

An Exploration of Real-Time Rhythmic Automatic Accompaniment

Tim Canfer

www.timcanfer.co.uk

Submitted in partial fulfilment of the requirements for the degree of [MMus, Music
Technology and Computer Music].

School of Music, University of Leeds, September 2010

Abstract

This dissertation is an exploration of technologies to enable real-time rhythmic automatic accompaniment: a system that can react to incoming live music and play back previously prepared accompaniment in time. The main areas of study that will be needed to enable such a system are beat tracking and score following. Accordingly, the majority of this dissertation is a literature survey, evaluation and discussion of beat tracking, tempo induction and score following technologies, focusing on the largely overlooked area of live, real-time beat tracking.

The problem of context and categorisation of live, real-time beat tracking as a field of study is discussed. Typically beat tracking is exclusively a Musical Information Retrieval (MIR) subject, however this is only a fair method of evaluating offline beat tracking. It is argued here that live beat tracking in this context is really a Human Computer Interaction (HCI) study, as it is the music that is being used to control a computer system.

The use and drawbacks of backing tracks in popular music is discussed and automatic accompaniment is suggested as a more musical use of live music technology. The lack of real-time rhythmic automatic accompaniment in popular music is explored and a practical implementation of real-time rhythmic automatic accompaniment for popular music is planned.

Table of Contents

1. Introduction	6
1.1 Aims and Objectives	6
1.2 Motivation	6
2. Discussions of Terms	8
2.1 Title Terms	8
2.2 MIR Terms	9
2.3 Music Psychology Terms	12
3. Beat Tracking and Score Following in Context	17
3.1 A Brief History of Beat Tracking and Score Following	17
3.2 Applications of Beat Tracking	20
3.3 MIR and Evaluations	23
4. Real-Time Rhythmic Automatic Accompaniments	27
4.1 Existing Beat Trackers and Score Followers	27
4.1.1 Beat Tracking	27
4.1.1.1 Academic Beat Trackers	28
4.1.1.2 Commercial Beat Tracker	33
4.1.2 Score Following	34
4.1.2.1 Academic Score Followers	34
4.1.2.2 Commercial Score Followers	36
4.2 Why is Real-Time Rhythmic Automatic Accompaniment for Popular Music so Rare?	37
4.3 Classification of Real-Time Automatic Accompaniment for Popular Music	39

5. Suggested Implementation	41
5.1 System Overview	41
5.1.1 Cue Responsive Beat Tracking	41
5.1.2 Flexible Score Following	42
5.2 System Elements	44
5.2.1 System Outline	44
5.2.2 Audio Inputs	44
5.2.3 Control Inputs	45
5.2.3.1 Timing Control	45
5.2.3.2 Direct Cues	46
5.2.4 Beat Trackers	48
5.2.5 Control Sequencer	49
5.2.6 Sequencer	49
5.2.7 Computational Load	51
6. Conclusion	52
Bibliography	53

Appendix

Appendix Forum Communication	58
B1. Cycling '74 > Max For Live Forum	58
B2. Reaper Forums > Reaper General Discussion Forum	62
B3. Reaper Forums > Reaper Development Forum	64

List of Figures

Figure 1. B-Keeper MaxMSP Screen Shot	29
Figure 2. btrack~ MaxMSP Help File Screen Shot	31
Figure 3. beat~ MaxMSP Help File Screen Shot	32
Figure 4. Basic System Elements	44
Figure 5. Overall Design of Control Element Components	47

1. Introduction

1.1 Aims and Objectives

The main aims of this dissertation are to:

- produce work that promotes a more musical and creative use of technology for popular music; and
- work towards producing an effective and practical system of real-time rhythmic automatic accompaniment for specific live popular music scenarios.

The primary objectives of this dissertation are to:

- explore the history of beat tracking and score following and the current state-of-the-art-beat trackers and score followers, focusing on the application of real-time rhythmic automatic accompaniment;
- review and evaluate the literature of the major areas of influence;
- discuss the position of real-time rhythmic automatic accompaniment relative to all the academic disciplines that inform it;
- suggest and discuss an implementation of a specific system of real-time rhythmic automatic accompaniment.

1.2 Motivation

The intention of this project is to work towards creating a real-time rhythmic automatic accompaniment system, designed specifically for popular music. To expand, this is a performance system designed to run on a laptop on stage, which is capable of reacting to live music then synchronising and playing back accompaniment. An analogy for the system is that of another musician, one that can trigger and manipulate a massive array of instruments and sounds and can intuitively play *with* the other musician(s).

The motivation for this project has come from my personal use of backing tracks and discussions with other musicians who have used backing tracks. These are pre-recorded stereo audio tracks created to accompany musicians (generally .wav or .mp3 files). Accepted best practice is to set up a click track or monitor mix in the left channel that is sent to the monitors and to set up a mono mix in the right channel that is sent to

the front of house speakers.¹ The reasons for this split are: the need for a count in (or some other kind of guide that is unheard by the audience), the need for an even mix and some sound reinforcement systems are mono.

The advantages of playing with a backing track are apparent, especially for anyone with experience of how many instrument tracks are often used in the studio. It is common to add many elements to the recording of a popular song that would be impractical to perform live (for example strings, extra percussion and sound effects). The backing track allows musicians to recreate that sound, or add different, more experimental elements. The disadvantages of playing with a backing track, while less apparent, are clear to any musician who enjoys the freedom of expression and interaction of playing with other musicians who will clearly play a subtly (or sometimes blatantly), different version of the music every time. This is due largely to changes in tempo and subtle changes in timing and character of each note. The backing track restricts this freedom, locking the tempo to a pre-prescribed and never changing track, inevitably causing a more mechanical style of playing.

Many musicians (again, myself included) learn to work around the restrictions of the backing track technology because of the advantages it offers, but more commonly, they abandon the concept, for the reasons stated above.

This project is an attempt to create a more musical backing track. The proposed system is essentially a reactive, sequenced backing track, which could even prove to be interactive if it in turn has any influence on the musicians playing.

¹ Gavin Harrison, 'Preparing Backing Tracks for Live Use', *Sound on Sound*, November 2003, <<http://www.soundonsound.com/sos/nov03/articles/backingtracks.htm>> [accessed 3 May 2010].

2. Discussion of Terms

This project covers a wide range of academic disciplines, from Music Technology, Computer Music and Music Psychology to Machine Learning, Signal Processing, MIR and HCI. Because of this it is necessary to offer a clear definition of the terms used in the context of this project. The areas for terms that are necessary to clarify are Title terms, MIR terms and Music Psychology terms.

2.1 Title Terms

Automatic Accompaniment

While the terms ‘automatic accompaniment’ and ‘automatic musical accompaniment’ have been used by Davies and Plumbley to describe a similar type of system outlined in section 1.2, the term ‘automatic accompaniment’ is most commonly used to describe the offline process of analysing an incoming melody and generating chords to accompany it.² Software such as ‘MySong’ or ‘Band in a Box’ have made this popular for people without the musical training who want to write songs, as it offers quick and predictably consonant results.³ Amusingly, after the release of this software it became popular to process well-known popular melodies through the software and post the generally trite results onto YouTube.

² Matthew E. P. Davies, Paul M. Brossier and Mark D. Plumbley, ‘Beat Tracking Towards Automatic Musical Accompaniment’, *Proceedings of the Audio Engineering Society Convention*, (2005), (p. 1).

³ Ian Simon, Dan Morris and Sumit Basu, ‘MySong: Automatic Accompaniment Generation for Vocal Melodies’, *Proceedings of the ACM Conference on Human Factors in Computing Systems*, (2008) <<http://research.microsoft.com/en-us/um/.../mysong/mysongchi2008.pdf>> [accessed 22 July 2010].

Rhythmic Automatic Accompaniment

The term 'rhythmic automatic accompaniment' is used for a system that tracks the beats of music to generate time information, which is then used to produce rhythmic accompaniment.⁴ In the system proposed here this information could be used to play either percussive or pitched accompaniment, but as they would be pre-prepared MIDI events only the time at which they occur would be variable, so both pitched or percussive sounds are effectively being treated as rhythmic accompaniment.

Real-Time Rhythmic Automatic Accompaniment

This dissertation is concerned with a system that is created for a live performance need, so it is necessary to be accurate that the system discussed is clearly understood to be 'real-time', i.e. at the time the music is being played, otherwise it really defeats the point of the project. The difference between real-time and offline systems is a very important one as, generally, offline systems have a totally different way of working to real-time systems.

2.2 MIR Terms

These are terms that are specific to MIR, an area of study that is still relatively young, so their meaning can vary greatly depending on context. This is particularly true of beat tracking, so much so that the next chapter of this literature review is called 'beat tracking in the context of real-time rhythmic automatic accompaniment'.

⁴ Matthew E. P. Davies, 'Towards Automatic Accompaniment', (unpublished doctoral thesis, University of London, Queen Mary, 2007), p. 1.

Beat Tracking

One of the main elements of real-time rhythmic automatic accompaniment is a technique referred to as beat tracking. As the name implies, beat tracking is the process of tracking beats within music, generally described as a computer program equivalent of foot tapping with a piece of music. However, this seemingly simple and intuitive task is not only particularly difficult to implement with a computer and so far, beat tracking algorithms are not able to perform as well as human beat trackers⁵, but it remains a challenge to accurately define.⁶ This is mainly due to the ambiguity of application; different applications require different types of beat tracking. It is worth noting that real-time rhythmic automatic accompaniment is one of the applications of beat tracking receiving the least amount of attention. The reasons for this are discussed in section 4.2.

The main difference in beat trackers arises from there being different beats that one can track; either perceived beats or performed beats. (This is discussed further in section 3.2.) The general consensus is that beat tracking is as Davies and others (2009) state: “The aim of a beat tracker is to recover a sequence of time instants from a musical input that are consistent with the times when a human might tap their foot”,⁷ that is, perceived beats. However, an alternate definition is offered from the same institution (Queen Mary, London) by Dixon (2007): “Beat tracking is the task of identifying and synchronising with the basic rhythmic pulse of a piece of music”, that is, performed beats.⁸ The reason for this discrepancy is that the Dixon system is designed primarily for performance analysis and the Davies system is working towards real-time rhythmic automatic accompaniment (this is discussed further in section 3.2) but not referring to or explaining this does not help clarity in the area.

⁵ Martin F. McKinney and others, ‘Evaluation of Audio Beat Tracking and Music Tempo Extraction Algorithms’, *Journal of New Music Research*, 36:1 (2007), 1–16 (p. 10).

⁶ Simon Dixon, ‘Automatic Extraction of Tempo and Beat from Expressive Performances’, *Journal of New Music Research*, 30:1 (2001), 39–58 (p. 39).

⁷ Matthew E. P. Davies, Norberto Degara and Mark D. Plumbley, ‘Evaluation Methods for Musical Audio Beat Tracking Algorithms’, *Technical Report C4DM-TR-09-06*, Queen Mary University of London, Centre for Digital Music, (2009)

<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.152.6936&rep=rep1&type=pdf> [accessed 31 March 2010] (p. 1).

⁸ Simon Dixon, ‘Evaluation of the Audio Beat Tracking System Beatroot’, *Journal of New Musical Research*, 36:1 (2007), 39–50 (p. 1).

Tempo Extraction and Tempo Induction

Beat tracking is very closely related to tempo extraction, that is, determining the tempo of a piece of music. McKinney and others (2007) succinctly describe the difference thusly: “Tempo extraction aims at determining the global speed or tempo of a piece of music, while beat tracking attempts to locate each individual beat”.⁹

Tempo induction is essentially the same process as tempo extraction, and authors will mainly stick to one or the other term, but there is some confusion. For example, in Pikrakis and Theodoridis (2007), although tempo induction is used in the title, both tempo induction and tempo extraction are used twice each on the first page, with no clear reason for the alternate terms.¹⁰

The lack of a standard consensus on this and other terms perhaps indicates how much tempo extraction and beat tracking are new areas of study. In correspondence, Matthew E. P. Davies suggests that induction implies a human process and extraction is any means to find the tempo (*all private correspondence has been removed from the public version of this dissertation*). This seems a reasonably common sense approach and because it seems to be the most widely used term and for the sake of clarity the term tempo extraction will be used in this dissertation.

Driving Function Generator and Periodicity Detector

The two main elements for any audio beat tracking or tempo extraction algorithm can be considered the driving function and the periodicity detector. While this is somewhat general and there are several algorithms that blur these boundaries, it is useful to define these elements as there are many differences and similarities between these two elements.

The driving function generator is the element that processes the incoming audio and generates a driving function, which is then used by the periodicity detector to detect periodicities and then estimate the tempo and or beat times. There are several different

⁹ McKinney and others, ‘Evaluation of Audio Beat Tracking and Music Tempo Extraction Algorithms’, p. 1.

¹⁰ Aggelos Pikrakis and Sergios Theodoridis, ‘An Application of Empirical Mode Decomposition on Tempo Induction from Music Recordings’, *Proceedings of the International Society for Music Information Retrieval Conference*, (2007), (p. 1).

ways to considering beat tracking algorithms but these terms are chosen as they are the standard terms used in the 2006 Audio Beat Tracking task of the Music Information Retrieval Evaluation eXchange (MIREX).¹¹

Causal

The term ‘causal’, in the context of beat tracking, comes from signal processing (*all private correspondence has been removed from the public version of this dissertation*), and means that the system discussed is predictive, i.e. it attempts to predict when future beats will occur.¹² A predictive or causal system designed to smoothly change tempo in a way that mimics other musicians is necessary in a system to be used in a live environment as unexpected tempo changes are generally undesirable (see section 2.3 ‘Perceived Tempo’).

This term is an example of a clash of definitions that occurs occasionally with interdisciplinary subjects. Within signal processing causal is an accepted concept covering causal systems (past and present inputs only), non-causal systems (past, present and future inputs) and anti-causal systems (future inputs only). However, from an interdisciplinary perspective, there seems little reason to use the term causal as it comes laden with such a well-established definition in everyday life. At best it is only in the most tenuous interpretation of causality that can link the concepts. In most other disciplines the word causal is far easier to understand, i.e. something relating to or acting as a cause.

This can be an intense source of confusion when working with several disciplines and trying to maintain clear understandings of the terms and concepts.

2.3 Music Psychology Terms

The importance of understanding how the human brain interprets and processes rhythm is vital to beat tracking. While the standard meaning of many of the terms used here is

¹¹ McKinney and others, ‘Evaluation of Audio Beat Tracking and Music Tempo Extraction Algorithms’, p. 2.

¹² Davies, p. 19.

common knowledge to musicians (beat, bar, etc.) a specific definition that clarifies the terms used in context is beneficial to an understanding of the systems and evaluations discussed. Unsurprisingly, these terms concentrate on the various types of rhythmic elements in music and both highlight and stem from a major focus of beat tracking, that of metrical levels of timing.

Metrical Levels

The different metrical levels are the different rates at which different people will naturally tap along with the music.¹³ Musicians know this as half, double and triple time. An understanding of how metrical levels can vary is particularly important to beat tracking. The following quotation from McKinney and Moelants (2006) shows both the importance of relative tempi strength at different metrical levels and the difficulty of extracting this information from acoustic sources:

“it is not clear whether or not the relevant information on perceptual accents can be readily extracted from a complex acoustic waveform. A method to extract relevant accent information in order to better predict the relative salience of perceptual tempi for all individual musical excerpts would be a boon for systems that automatically extract tempo from musical audio signals. Previous attempts at automatic tempo extraction (typically measured against notated tempo or that annotated by a single person) have met with some success but the most common error is a tempo octave error (Alonso, David, & Richard, 2004; Goto & Muraoka, 1998; Klapuri, Eronen, & Astola, 2006; Scheirer, 1998). A model that could predict the relative strength of tempi at different metrical levels would be able to guide such systems to more accurately predict the perceived tempo.”¹⁴

Tactus and Beat

The beat is the most recognisable rhythmic term for any listener of music. It is described by Davies and Plumbley (2007) as “the rate at which humans are most likely to tap”, (this

¹³ Dixon, 'Automatic Extraction of Tempo and Beat from Expressive Performances', p. 51.

¹⁴ Martin F. McKinney and Dirk Moelants, 'Ambiguity in Tempo Perception: What Draws Listeners to Different Metrical Levels?', *Music Perception*, 24:2 (2006), 155–66 (p.156).

source is also quoted for the Tatum and the Measure sections below).¹⁵ In a bar of 4/4 music, the beat is the crotchet or the quarter note and the element we are attempting to extract as accurately as possible in beat tracking.

Tatum

The tatum is the one rhythmic term not immediately clear to a musician. Within beat tracking, the tatum is “the lowest or fastest metrical level”.¹⁶ While the beat may be prominent in some forms of music, there may be other elements such as divisions of the beat that are clearly heard. For example, there may be musical elements heard on the quaver or eighth note or semi-quaver or sixteenth note.

Bar and Measure

The term measure “indicates the grouping of beats into bars”.¹⁷ While clear to musicians, this is the grouping in metrical units of repeating rhythmic and melodic elements. The Oxford Companion to Music explains the terms in the context of notation, obviously stating that the bar is the vertical line on a staff, but that bar is now also the name for the metrical unit itself.¹⁸

1 ee and ah

A standard method of communicating syncopated rhythmic information for music with a 4/4 rhythm is by using the drummer sixteenth parlance of “1-ee-and-ah 2-ee-and-ah 3-ee-and-ah 4-ee-and-ah”: the numbers being the beats and the sounds being the different sixteenth positions between the beat. For example, a quaver rest followed by a quaver would be indicating a beat on the ‘1and’ and then a dotted quaver rest followed by a

¹⁵ Matthew E. P. Davies and Mark D. Plumbley, ‘Context-Dependent Beat Tracking of Musical Audio’, *IEEE Transactions on Audio, Speech, and Language Processing*, 15:3 (2007), 1009–20 (p. 1010).

¹⁶ Davies and Plumbley, p. 1010.

¹⁷ Davies and Plumbley, p. 1010.

¹⁸ ‘bar’, *The Oxford Companion to Music*, (2002).

semi-quaver would be indicating a beat on the 'Zah' (following on from each other). Using these terms makes it far easier to convey any syncopation in sixteenths.

Score Following

According to Jordanous and Smaill (2009) "Score following is the process whereby a musician follows another musician's playing of a musical piece, by tracking their progress through the score of that piece."¹⁹ The musician, in the case of real-time rhythmic automatic accompaniment, would be a computer. So instead of tracking the beats of music, the notes are tracked, normally using monophonic pitch detection. To do this, a score of the music is needed and this is generally in a MIDI format for computers to score track.

Notated Tempo, Score Tempo and Performed Tempo

Notated tempo is the tempo as written on the score and played by the performer. For the sake of this project the notated tempo is assumed to be the same as performed tempo. It is clear that the tempo of any expressive performance of a piece of music will differ from one performer to the next, but while this will only be rigidly fixed if played with a click track (as with much popular music in the studio), there will be a tempo term and or beats per minute (bpm) specified for most performed music, which will generally stay to this tempo, or change where indicated.

Perceived Tempo

For a system of real-time rhythmic automatic accompaniment to be effective, whether it uses score following or beat tracking, it must react in a way that is close enough to how a human musician would react. To achieve this, an understanding of the distinction between notated tempo (or score tempo) and perceived tempo is important.

¹⁹ Anna Jordanous and Alan Smaill, 'Investigating the Role of Score Following in Automatic Musical Accompaniment', *Journal of New Music Research*, 38:2 (2009), p. 197.

The perceived tempo is how the listener experiences tempo. It is the difference between notated and perceived tempo that is interesting and is directly relevant to real-time rhythmic automatic accompaniment. In an experiment to discover perceptual smoothness of tempo changes, Dixon and others (2006) discovered that generally the participants responded to tempo changes on the beat after the change occurs.²⁰ As the title suggests, they also discovered both a significant smoothing of perceived tempo changes compared to performed tempo in a tapping experiment and a preference for a click track that was more smoothed (i.e. less accurate to the actual note occurrences) in a listening experiment.

This is a very general rule and stops applying when listeners have a prior knowledge of the piece and the musicians who are playing that piece. The ability for musicians to synchronise with each other's timing variations will be familiar to anyone who has played with other musicians and become used to their style of playing. Shaffer (1984) used piano duet studies to confirm this and this study is further clarified by Appleton, Windsor and Clarke (1996) to show that musicians are able to synchronise more under conditions of visual feedback (being able to see each other when playing). This demonstrates the importance of sight as well as sound for a truly musical and interactive system of real-time rhythmic automatic accompaniment. It would therefore be desirable to provide a Graphical User Interface (GUI) showing useful musical feedback.

A common problem with evaluating beat tracking algorithms is that they are tested against ground truth tapped tempo, i.e. perceived tempo, but they are often not designed to allow for this difference.

²⁰ Simon Dixon, Werner Goebel and Emiliios Cambouropoulos, 'Perceptual Smoothness of Tempo in Expressively Performed Music', *Music Perception*, 23:3 (2006), 195–214 (p. 210).

3. Beat Tracking and Score Following in Context

As beat tracking is the main element in a system of real-time automatic accompaniment, a wider understanding of the relevant academic areas is important. This chapter discusses the history, various applications, place in academia and methods of evaluating different systems of beat tracking in order to see its place within the fields it occupies, namely: music, music psychology, signal processing and computing.

3.1 A Brief History of Beat Tracking and Score Following

The most basic way of categorising beat tracking and score following techniques is by the type of input, either Information or Audio. This is a very general and blunt way of considering beat tracking and while there are several exceptions to these categories (several multiple agent and probabilistic algorithms can accept MIDI input as well as Audio) it gives a rough chronological order to the methods and allows a useful overview of the different types of systems.

Information

The earliest systems used symbolic information, specifically note onset times derived from a score, generally intended to supply psychological models. For example, the 1971 Longuet-Higgins and Steedman study of pulse detection in Bach's fugues²¹ can be seen as an early precursor. This led to various rule-based systems as there was not yet the processing power available for complex algorithms: for example, the 1994 Parncutt perceptual model of pulse salience, which included psychoacoustic factors to improve accuracy.²²

²¹ H. Christopher Longuet-Higgins and Mark J. Steedman, 'On Interpreting Bach', in *Machine Intelligence 6*, ed. by Bernard Meltzer and Donald Michie (Edinburgh: Edinburgh University Press, 1971), pp. 221–41.

²² Richard Parncutt, 'A Perceptual Model of Pulse Salience and Metrical Accent in Musical Rhythms', *Music Perception*, 11:4 (1994), 409–64.

An alternative method to rule-based approaches was proposed in 1992 by Miller and others, using oscillator resonance. This method used a bank of oscillators responding to the incoming signal with the oscillator resonating most indicating the dominant periodicity.²³ This method is based heavily on psychological data, in mimicking how human behaviour works and is particularly suited to causal analysis,²⁴ so theoretically this could be a useful system for real-time rhythmic automatic accompaniment.

As a particularly robust standard for performance data, MIDI is used and well suited for tempo extraction and beat tracking and is far simpler than having to extract the important note onsets from an audio stream. Systems such as Block and Dannenberg 1985²⁵ and Bryson 1995²⁶ used MIDI and it is often used as an alternative driving function to algorithms that also beat track from audio signals (for example Dixon 2001²⁷ and Raphael 1999²⁸).

Audio

The first published attempts to track an audio source came in 1984 from Vercoe²⁹ and Dannenberg.³⁰ Both articles work towards systems of score following to enable musical accompaniment and use monophonic pitch detection (the Vercoe system also using optical sensors to increase accuracy). The Vercoe article is particularly worthy of note as

²³ McKinney and others, p. 2.

²⁴ Simon Hainsworth, 'Techniques for the Automated Analysis of Musical Audio', (unpublished doctoral thesis, University of Cambridge, 2003), p. 64.

²⁵ Joshua J. Bloch and Roger B. Dannenberg, 'Real-time Computer Accompaniment of Keyboard Performances', *Proceedings of International Computer Music Conference*, (1985), 279–89.

²⁶ Jonna Bryson, 'The Reactive Accompanist: Adaptation and Behaviour Decomposition in a Music System', in *The Biology and Technology of Intelligent Autonomous Agents*, ed. by Luc Steels (Heidelberg: Springer-Verlag, 1995), pp. 365–76.

²⁷ Dixon, 'Automatic Extraction of Tempo and Beat from Expressive Performances', p. 39.

²⁸ Christopher Raphael, 'Automatic Segmentation of Acoustic Musical Signals Using Hidden Markov Models', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21:4 (1999), 360–70.

²⁹ Barry Vercoe, 'The Synthetic Performer in the Context of Live Performance', *Proceedings of the International Computer Music Conference*, (1984), 199–200.

³⁰ Roger B. Dannenberg, 'An On-Line Algorithm for Real-Time Accompaniment', *Proceedings of the International Computer Music Conference*, (1984), 193–98.

he calls his real-time rhythmic automatic accompaniment system the 'Synthetic Performer'.³¹ Interestingly, the goal stated in that paper is to "replace any member of the group by a synthetic performer (i.e. a computer model) so that the remaining live members cannot tell the difference", which is ambitious even with the processing power available today.

A general shift in the different fields of academic research is particularly true, as noted by Hainsworth (2003) in discussing a literature survey of beat tracking "The early work was undertaken in the fields of music perception and computer science though the emphasis has shifted towards engineering as computing power increased."³² This trend has continued apace and forms a significant consideration for this dissertation (see section 3.3).

While these studies mark the first attempt to beat track with an audio signal it was not until the 1994 Goto and Muraoka study that good results were found using a multiple agent approach and focusing on specific genres of popular music with reliably constant rhythmic characteristics.³³ More general approaches using a multiple agent approach have continued, most notably with the 2001 Dixon system Beatroot. Initially this was designed for a MIDI input, but an energy based onset detection front end enabled beat tracking of audio (although understandably less accurate).³⁴ In 2007 Dixon refined this design and made a Java version of Beatroot available and used spectral flux to gain a far more accurate driving function.³⁵

Interestingly, one of the first of the general approaches to beat tracking came in 1998 from Scheirer,³⁶ who used oscillator resonance to create a causal system that while fairly successful suffered from a long burn in period and a tendency to switch between metrical levels and subdivisions of the tactus.

Through the late 1990s and the first decade of the millennium, the complexity of algorithms has increased massively in the area of signal processing due to the availability of far more processing power. As studies continued, several large scale

³¹ Vercoe, p. 199.

³² Hainsworth, p. 46.

³³ Masataka Goto and Yoichi Muraoka, 'A Beat Tracking System for Acoustic Signals of Music', *Proceedings of the Second ACM International Conference on Multimedia*, (1994), 365–72.

³⁴ Dixon, 'Automatic Extraction of Tempo and Beat from Expressive Performances', p. 39.

³⁵ Dixon, 'Evaluation of the Audio Beat Tracking System Beatroot', p. 39.

³⁶ Eric D. Scheirer, 'Tempo and Beat Analysis of Acoustical Musical Signals', *The Journal of the Acoustical Society of America*, 103:1 (1998), 588–601.

evaluations of tempo extraction or beat tracking have taken place: Firstly, the 2004 tempo extraction contest organised during the International Conference on Music Information Retrieval (ISMIR), which was summarised and reported on in the Institute of Electrical and Electronics Engineers (IEEE) Transactions on Audio Speech and Language Processing. (Quite an indication of how much engineering has a part to play in modern tempo extraction/beat tracking). The other evaluations are discussed in section 3.3. State-of-the-art beat trackers and score followers are discussed in section 4.1.

3.2 Applications of Beat Tracking

The applications of score following are relatively well defined, being mainly for accompaniment or performance analysis. The only real exception to this is the offshoot application of offline score following, which allows re-synthesis of a pre-recorded performance so that it can be used for variable speed accompaniment.³⁷

In contrast, the applications of beat tracking are numerous and poorly defined. A common problem with more recent studies, in particular the MIREX evaluations, is that beat tracking has become an area of research in its own right and is often explored without context, the implication being that the systems being evaluated would be suitable for all applications. This is obviously not the case; there is a need for a more accurate structure to put the different types of beat tracking systems into the context of their respective applications. As Dixon (2001) states, these applications fall under the categories of: performance analysis, perceptual modelling, audio content analysis, and synchronising with a musical performance, (under which falls real-time rhythmic automatic accompaniment).³⁸

³⁷ Christopher Raphael, 'A Hybrid Graphical Model for Aligning Polyphonic Audio with Musical Scores', *Proceedings of the International Society of Music Information Retrieval*, (2004), (p. 1).

³⁸ Dixon, 'Automatic Extraction of Tempo and Beat from Expressive Performances', p. 39.

Performance Analysis

As discussed in section 3.1, performance analysis is the earliest application of beat tracking as it is the least computationally intensive application. This acted (and acts) as a solid source of information to accelerate performance analysis, giving insight into emotional and structural characteristics of the performance. What is needed for this application is to track performed beats (rather than perceived beats, see section 2.3 'Perceived Tempo'). Notably, the Dixon system Beatroot is designed to track performed beats as it is designed principally for performance analysis. It is also non-causal (offline) and allows a graphical representation of where/when beats occur for more in-depth analysis.

Perceptual Modelling

The ability to analyse musical performances has happened alongside a number of studies into perceptual models of rhythm in music.³⁹ It is worth noting that while ideally these areas of study should be inherently multi-disciplinary, the music psychology articles rarely draw upon influences outside of music psychology. For example, many articles written by academics in the fields of engineering and music reference, like Dixon Goebel and Cambouropoulos (2006), draw upon studies of rhythmic perception from the world of music psychology.⁴⁰ However, in music psychology articles about perceptual modelling of rhythm in music there are few studies with references to similar studies from engineering that could inform and enrich these studies, i.e. Repp and Keller (2010),⁴¹ although a notable exception is the excellent Desain and Honing (1993) article 'Tempo Curves Considered Harmful'.⁴²

³⁹ Bruno H. Repp, 'Sensorimotor Synchronization: A Review of the Tapping Literature', *Psychonomic Bulletin & Review*, 12:6 (2005), 969–92.

⁴⁰ Dixon, Goebel and Cambouropoulos, p. 196.

⁴¹ Bruno H. Repp and Peter E. Keller, 'Self Versus Other in Piano Performance: Detectability of Timing Perturbations Depends on Personal Playing Style', *Experimental Brain Research*, 202 (2010), 101–10.

⁴² Peter Desain and Henkjan Honing, 'Tempo Curves Considered Harmful', *Contemporary Music Review*, 7 (1993), 123–38.

Audio Content Analysis

Audio content analysis allows the automatic categorising and indexing of music in libraries and databases. Because of this, it has the potential to be the most widely used category of applications for consumer and industry environments. Tempo extraction is already a function of several computer programs intended for the ID tagging of digital audio, such as the Native Instruments DJ software Traktor (called beats per minute (bpm) detection). While the tempo extraction method of Traktor is not discussed, it appears to be a simple energy transient detection method, but is not reported to be particularly effective, even with music that has a straight four to the floor beat.

An accurate and reliable algorithm for ID tagging of tempo could quite easily be a function of a media player such as iTunes in the near future, to enable effective playlisting and musical retrieval. This is a typical example of an MIR task and is evaluated on a wide scale by MIREX, as well as other tasks with clear ID tagging applications such as audio genre classification and audio artist information.

A more academic and bespoke scenario for audio content analysis is the software MusicPlan developed by Evans 2007, which involves the creation and annotation of boundary points of music to assist in the planning of televised musical performances.⁴³ There is also a strong case made for the segmentation of music using offline beat tracking so that the units of time are measured in bars and beats.

Both tempo extraction for ID tagging and boundary point creation for either synchronisation or editing are offline tasks, and would ideally extract the perceived tempo. This is the case for the majority of audio content analysis applications as they generally exist to supply information about digital audio files.

Synchronising with a Musical Performance

This is the only exclusively real-time category of applications and covers several different implementations of similar systems. A particular application with clear potential for music processing devices is the synchronisation of audio effects using a low frequency oscillator linked to the beats of the music. A good example of this is the Stark,

⁴³ Michael J. Evans, 'Interactive Beat Tracking for Assisted Annotation of Percussive Music', *Proceedings of the Audio Engineering Society Convention*, (2007), (p. 10).

Plumbley and Davies (2007) study where the Davies and Plumbley causal beat tracking algorithm was successfully applied to automatically synchronise time variant effects such as tremolo, flange and auto-wah.⁴⁴

The other main application of beat tracking within this category is the focus of this dissertation, that is, for the use of real-time musical accompaniment. While initially a computer supplying appropriate accompaniment to music would seem a fairly simple concept, the reality is far more splintered, with many differing terms and approaches based on wildly different motivations and goals. In particular, most systems either assume or work from the perspective of real-time score following. This, of course, assumes that the musical score is known beforehand and is conveyed in some way to the system. There is also the question of who or how many musicians are being 'followed' and in what way.

Other methods of synchronising with a musical performance involve 'on the fly' accompaniment, i.e. without a score. This would be achieved with a system of beat tracking and a more generative and textural means of supplying appropriate accompaniment.

In academic study in particular this is an area of study that has not had sufficient consideration to merit well-defined areas of its own. A major aim of this dissertation is to apply some order to the various studies and systems and to discuss both the advances made and the pitfalls and drawbacks of real-time rhythmic musical accompaniment.

3.3 MIR and Evaluations

Beat tracking and the various applications listed above all come under the academic umbrella of Music Information Retrieval (MIR). As an offshoot of Information Retrieval, a recent field in its own right, MIR is a very new field of study. Music Information Retrieval is a field of study born from advances in computer technology allowing the kind of processing that can drive these analyses.

The yearly conference for MIR is ISMIR (the International Symposium on Music Information Retrieval), which started in 2000 as a development from smaller music IR

⁴⁴ Adam M. Stark and Mark D. Plumbley and Matthew E. P. Davies, 'Audio Effects for Real-Time Performance Using Beat Tracking', *Proceedings of the Audio Engineering Society Convention*, (2007).

groups formed as a part of the ACM (Advanced Computer Machinery) Digital Library Conference 1999 and SIGIR (Special Interest Group on Information Retrieval) 1999 and the OMRAS (Online Music Recognition and Searching) Project.⁴⁵

Formal evaluation of MIR algorithms is conducted by The Music Information Retrieval Evaluation eXchange (MIREX), described by Dowie (2008) as "the community-based framework for the formal evaluation of Music Information Retrieval (MIR) systems and algorithms."⁴⁶

MIREX grew from the Audio Description Contest in ISMIR 2004 and was inaugurated in 2005. Looking at this background it is clear to see a huge predominance of engineering, specifically in the domain of signal processing, but the drawback of this is the perception that music-specific knowledge is less of a requirement. This predominance of an engineering approach is understandable in a wider context, considering the near ubiquitous use of digital audio to record, process, store and distribute music in the twenty-first century and perhaps a symptom of the continuing invasion of technology on the arts. For musicians, especially popular musicians, the effect of this movement is that there is a serious disadvantage in a lack of awareness of the technology used to make and store music: from the recording horns of the 1910s to the laptop orchestras of the 2000s.

On the other side of the academic fence however, engineers working with music are also enormously disadvantaged if there is no real understanding of the application of their work. The clear conclusion is that a lack of musical expertise in MIR would have massive repercussions on the relevance and validity of research and while this is not yet the case, there is a very real risk of this happening.

There are generally around ten tasks a year that are evaluated by MIREX.⁴⁷ The tasks that are of specific interest for this dissertation are Audio Beat Tracking, Audio Tempo Extraction and Score Following.

The beat tracking and tempo extraction tasks are very similar but were evaluated separately. A beat tracking task was run in 2006 and 2009, and a tempo extraction task

⁴⁵ Donald Byrd and Michael Fingerhut, 'The History of ISMIR – A Short Happy Tale', *D-Lib Magazine*, 8:11 (2002) <<http://www.dlib.org/dlib/november02/11inbrief.html#BYRD>> [accessed 9 August 2010].

⁴⁶ Stephen J. Downie, 'The Music Information Retrieval Evaluation Exchange (2005–2007): A Window into Music Information Retrieval Research', *Acoustical Science and Technology*, 24:4 (2008), 247–55 (p. 247).

⁴⁷ The MIREX tasks and results can all be seen at the MIREX wiki, which acts as hub for all relevant communications <http://www.music-ir.org/mirex/wiki/MIREX_HOME> [accessed 10 August 2010].

was run in 2005 and 2006. (There was also a well-documented competition as a part of ISMIR 2004, mentioned in section 3.1, entitled the ‘tempo induction contest’, but there is no mention about the change of terms.)

For tempo extraction, the difference between notated and perceived tempo is mentioned, but as there is no access to notated tempo it ends up that all algorithms are tested against perceived tempo.

The evaluation of the MIREX algorithms is based on a P-Score, ‘P’ standing for performance. For tempo extraction this is calculated by identifying the two most salient tempos for each musical extract giving each a figure (0–1) for salience. The algorithm’s ability to extract each tempo within 8 per cent accuracy was combined with the salience weighting to generate the P-Score (0-1).

For the beat tracking algorithms, the P-Score is calculated by generating an impulse train for each annotator and cross correlating with the output of the algorithm to see if the beat was within an error window of 1/5 of the beat.

The Score Following tasks were run in 2006 and 2008 and used a percentage precision score for each piece (58 files), and overall, which were calculated by taking the total number of events and taking away any missed notes.

There is much understandable importance placed on the way any system or algorithm is evaluated as any bias will have a huge impact on the outcome, and a large part of the MIREX wiki is spent discussing this. However there is a sense of the evaluations taking place in an academic vacuum, where application of these algorithms is secondary to the inner workings. While the evaluations are sound and admirably peer reviewed, the wiki seems somewhat circular and self-serving, as there are no examples of the algorithms being successfully (or unsuccessfully) applied. It is only in occasional articles where the evaluations are rounded up that the purpose of these applications is mentioned, for example, McKinney (2007).⁴⁸

Another interesting point to note is that although the beat tracking tasks have a range of genres spanning popular and art/classical music, there is no suggestion of any popular music being evaluated in the Score Following tasks. This is quite strange as the world of popular music has embraced technology both in its mainstream and particularly so in its fringes, in stark contrast to classical or art music. While scores are not used anywhere near to the same extent in popular music, there is a significant amount of

⁴⁸ Martin F. McKinney and others, ‘Evaluation of Audio Beat Tracking and Music Tempo Extraction Algorithms’, p. 2.

standard popular scores that could be used. This is certainly the case in commercial music instrument tuition software such as Smart Music, discussed in section 4.1.2.2.

4. Real-Time Rhythmic Automatic Accompaniments

4.1 Existing Beat Trackers and Score Followers

The following two sections discuss academic and commercial systems available, firstly for beat tracking and secondly for score following. The applications that have been chosen are ones that are relevant to this project; for example off-line beat trackers like Beatroot have been largely left out as they are of no use for Real-Time Rhythmic Automatic Accompaniment. The types of system that are of use are examples of causal software that can either beat track or score follow.

The systems have been separated into academic and commercial areas. Unfortunately, in the case of several commercial systems the workings are generally unknown, as the manufacturer keeps them secret. However, it is possible to surmise the basic methods.

As this is an academic Masters dissertation the main sources are academic so this is by no means an exhaustive list of commercial systems. There are a huge number of open source, shareware, freeware or projects available, which are impossible to cover in a document such as this. It is also worth noting that while all efforts have been made to find all important and relevant academic works, this is still a relatively young, sprawling and unregulated area of study making consistent research particularly difficult so it is possible that omissions may have been made.

4.1.1 Beat Tracking

The systems below are not separated into genre type as they are all designed for popular music, but they could be tailored to work with some types of art/classical music, although any music without predominant and repetitive percussion (such as most types of art/classical music) is inherently hard to beat track.

4.1.1.1 Academic Beat Trackers

While there has been much notable written work towards systems of causal beat trackers, for example the Davies (2007) thesis and several causal beat tracking algorithms, for example Goto (2001),⁴⁹ there are at present only two notable beat trackers that are readily available, up to date and run through a widely used application on a general platform (mac OS or Windows). (It is also worth noting that both of these beat trackers come from the Queen Mary University of London Centre for Digital Music.)

These two beat trackers are B-Keeper, Andrew Robertson's Max MSP application designed to drive Ableton Live, and btrack~ Adam Stark's Max MSP object.

Other beat trackers worth mentioning are the two Max MSP objects beat~ and op.beatitude~, and DrumTrack, a UGen element for use within Super Collider. That none of these beat trackers can run as an independent standalone system is another indication of both how new and how under resourced real-time rhythmic automatic accompaniment is. This is discussed further in section 4.2.

B-Keeper

B-Keeper is a standalone application made with Max MSP by Andrew Robertson at Queen Mary College, University of London. It is designed to drive the tempo of the live sequencer Ableton Live, so used alone it simply generates a MIDI click.

B-Keeper is the audio driven program with the closest aims to the ones set out in this dissertation, as it is developed specifically to do away with the reliance on a fixed click track for contemporary rock music. What is particularly interesting is that beat tracking in this environment is categorised by Robertson and Plumbley (2007) as Human Computer Interaction rather than Musical Information Retrieval.⁵⁰ The system was performed and demonstrated at the Live Algorithms for Music (LAM) Conference 2006 and the paper was published as a part of the New Instruments for Musical Expression Conference 2007. This really sets the system aside from the other beat trackers

⁴⁹ Masataka Goto, 'An Audio-Based Real-Time Beat Tracking System for Music with or without Drum Sounds', *Journal of New Musical Research*, 30:2 (2001), 159–71.

⁵⁰ Andrew Robertson and Mark D. Plumbley, 'B-Keeper: A Beat-Tracker for Live Performance', *Proceedings of the New Interfaces for Musical Expression Conference*, (2007), 234–7 (p. 234)

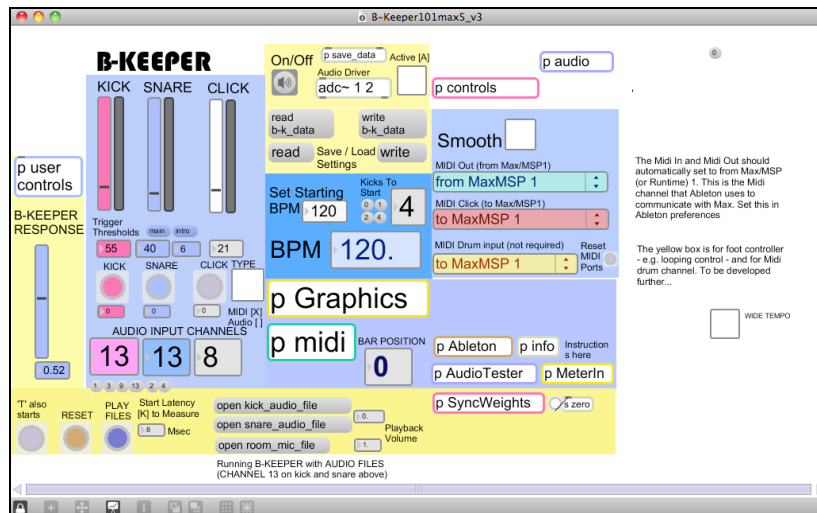
<www.elec.qmul.ac.uk/people/markp/2007/RobertsonPlumbley07-nime.pdf> [accessed 4 June 2010].

evaluated by MIREX in that it has a clear and specific application and has been made available for a whole community of musicians to use.

In its most recent form it takes the audio from the kick drum and snare drum. It then makes use of the bonk~ object developed by Miller Puckette, which employs spectral envelopes to detect the onsets of specific signatures. There is an excellent video of the LAM 2006 demonstration that clearly shows the system smoothly speeding up and slowing down.⁵¹

Interestingly, the system is designed to be overlooked by a user and there is a nudge function built in case the system goes out of phase, i.e. jumps to the off beat. It is even mentioned in the Robertson (2007) article that this happens in the LAM demonstration. This clearly takes some of the 'automatic' out of the accompaniment and shows that the system is prone to this kind of error. However the fix is a pragmatic one, especially when considering everything that could go wrong in the live environment of a rock gig. According to the communication with Andrew Robertson (*all private correspondence has been removed from the public version of this dissertation*), the more recent changes have made the B-Keeper more stable and there is a version created for Max for Live, which integrates deeper with the sequencer Ableton Live.

Figure 1. B-Keeper MaxMSP Screen Shot



⁵¹ This video can be seen at <http://www.elec.qmul.ac.uk/digitalmusic/downloads/b-keeper/LamConference2edit.mov> [accessed 13 August 2010].

btrack~

btrack~ is one of three Max MSP objects developed by Adam Stark and Matthew Davies, also from Queen Mary College, developed for beat synchronous harmonic analysis. This is a deceptively simple, audio in, bang on beat out object. The help file for btrack~ is shown in Figure 2.

The driving function for beat~ uses the complex spectral difference onset detection function.⁵² This sends onset information to the periodicity detector, which uses the dynamic programming approach used by the Ellis algorithm evaluated in the MIREX 2006 beat tracking task.⁵³ btrack~ is evaluated using a continuity-based evaluation metric discussed in Davies and Plumbley (2007).⁵⁴ Comparing this against the non-causal algorithms of Klapuri and Davies and Plumbley, btrack~ performs particularly well considering that a causal counterpart would be expected to fall significantly behind.

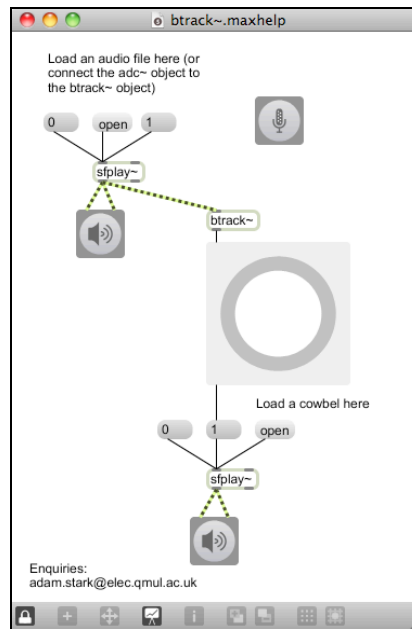
In a short, informal evaluation using a microphone and clicking fingers to a steady pulse it is surprisingly quick to react and tracked moderately well, but with a guitar it responded extremely well, adjusting quickly and tracking accurately, its performance far exceeding that of beatitude~. When tracking a song with a 'straight' beat, for example Billy Jean by Michael Jackson, btrack~ was following the beat solidly by the second beat. Using a song with strong syncopation and a textural instrumental introduction, for example Fresh Feeling by Eels, btrack~ took a bar of percussive sound before it started solidly tracking on either the beat, or the 'and' (see section 2.3 '1 ee and ah'). This is understandable as the song has a very strong accent on the 1and. The only real drawback is the lack of customising such as threshold, nudge function or any other means of tailoring the object to the audio.

⁵² Adam M. Stark, Matthew E. P. Davies and Mark D. Plumbley, 'Real-Time Beat-Synchronous Analysis of Musical Audio', *Proceedings of the International Conference on Digital Audio Effects*, (2009) <dafx09.como.polimi.it/proceedings/papers/paper_65.pdf> [accessed 13 August 2010].

⁵³ Martin F. McKinney and others, 'Evaluation of Audio Beat Tracking and Music Tempo Extraction Algorithms', p. 3.

⁵⁴ Davies and Plumbley, p. 1016.

Figure 2. btrack~ MaxMSP Help File Screen Shot



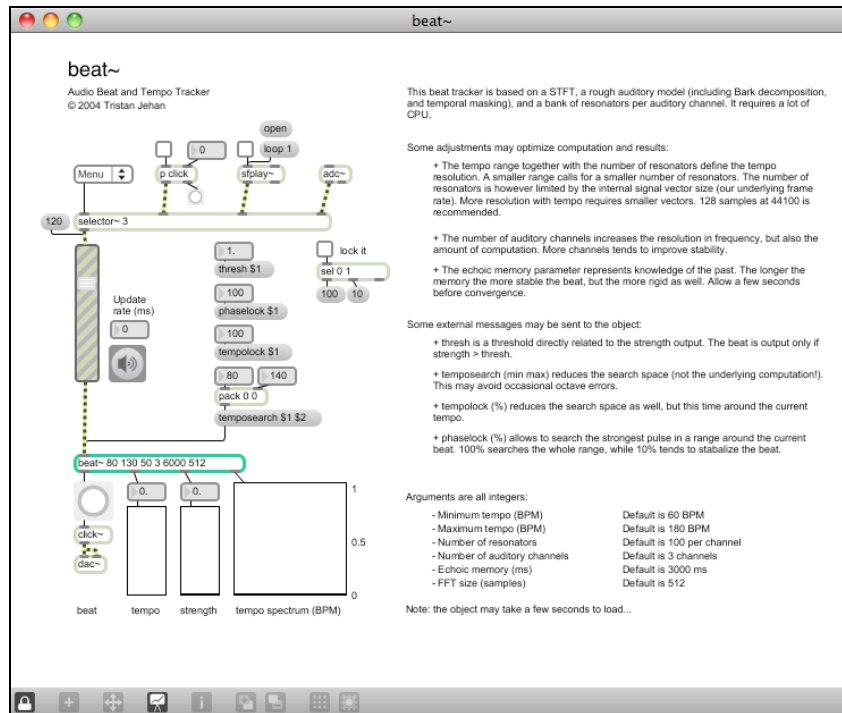
beat~

beat~ is a Max MSP object programmed by Tristan Jehan at Massachusetts Institute of Technology (MIT), and released to the Max community in 2004. The Max MSP help file for beat can be seen in Figure 3. beat~ works by using a short time Fourier transform that is warped into a bark scale to model the frequency response of the inner and outer ear. The frequency resolution is then lowered to 25 critical bands in order to reduce complexity.⁵⁵

Sadly, as is the case with many academic projects, there is little continued support as technology progresses. The versions of beat generally available are not optimised for Max 5 and I found that it would only work with a click track. Any attempt to drive the object with an audio signal, live or from a file, resulted in crashing and Max shutting down.

⁵⁵ Tristan Jehan, 'Downbeat Prediction by Listening and Learning', *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, (2005) <web.media.mit.edu/~tristan/Papers/WASPAA05_Tristan.pdf> [accessed 10 August 2010].

Figure 3. beat~ MaxMSP Help File Screen Shot



beatitude~

beatitude~ is a derivative of beat~ programmed by Olivier Pasquet in 1995 also using bark decomposition and a filter bank of 24 filters. While this works in Max 5 it does not perform very favourably.

In a short, informal evaluation using a microphone and clicking fingers to a steady pulse it was very slow to react and tracked either incorrectly or half time and double time more than the steady rhythm that was clicked. A single file with a strong and syncopated beat (Fresh Feeling by Eels) was tracked at the correct tempo for approximately half the time but the beat that it chose to click on was the off beat, presumably showing a tendency to track percussion sounds in the higher frequencies.

While this is clearly not a serious attempt at a proper evaluation, it does show that this beat tracker would not be of any real use in a system of real-time rhythmic automatic accompaniment.

DrumTrack

Nick Collins at Cambridge University created DrumTrack as a UGen element within Super Collider as a part of his BBCut2 (break-beat cut) software library. In the Collins 2005⁵⁶ article he evaluates the algorithm comparing it to the non-clausal Davies 2005 model and the causal Sheirer 1998 model. As one might expect, the non-causal algorithm outperforms DrumTrack, however the Sheirer algorithm falls well behind. While this is an indicator that DrumTrack performs relatively well it is definitely worth noting that the Sheirer model is, even in 2005, not state of the art. Another drawback of this system is that it is only really of use to experienced Super Collider users making it an extremely niche application.

4.1.1.2 Commercial Beat Tracker

There is only one system available commercially designed explicitly to beat track, however it is designed specifically for reactive accompaniment, a very similar rationale as explained in the motivation section 1.2 of this dissertation.

InTime

InTime, developed by Circular Logic, is a tempo tracking system that runs from a MIDI input and can act as either a MIDI sync master for a sequencer or it can play accompanying MIDI files on its own. It costs \$159 and was released in 2005. This effectively makes it a MIDI driven periodicity detector and its main strength seems to be the easy to use GUI, with various settings for sensitivity and smoothing.⁵⁷ It has the drawback of being MIDI driven so to be effectively driven by audio it would need an almost latency free audio to MIDI converter, or special MIDI triggers for drums (i.e. D-drums).

It should be noted that this is a fringe system, particularly esoteric and far from the reach of most musicians. The possibility of a system such as this developing into the

⁵⁶ Nick Collins, 'DrumTrack: Beat Induction from an Acoustic Drum Kit with Synchronised Scheduling', *Proceedings of The International Computer Music Conference*, (2005).

⁵⁷ Ingo Vauk, 'Circular Login InTime', *Sound on Sound*, June 2005

<<http://www.soundonsound.com/sos/jun05/articles/intime.htm>> [accessed 15 August 2010].

mainstream including audio inputs and become a feature of sequencers seems remote as InTime has not proved popular enough to warrant an update for Windows Vista or Windows7 platforms. However, that does not make real-time automatic accompaniment or the rationale behind it any less worthwhile.

4.1.2 Score Following

In the complete opposite way that beat tracking systems are generally designed for popular music, all of the following systems of score following are designed primarily for art/classical music. These systems are perhaps more likely to work with popular music because they simply need an instrument that can be followed and a score to follow.

The main reason for this genre divide must stem from the fact that popular musicians rarely, if ever, use scores. Of course popular music can be scored and will often be for session musicians, but this is contrary to the way that the music is played. Genres such as Blues and Jazz require a far more flexible means of recording music on paper, such as fakebooks.

4.1.2.1 Academic Score Followers

Score following in academia has received far greater attention and resources than beat tracking. This is shown in the focused approach to the score following team at IRCAM going back to the early work of Vercoe (see section 3.1), and the various fully developed systems mentioned below. The reason for this is that, in the vast majority, score following is the approach used to track real-time art/classical music and beat tracking is the approach used for popular music. This divide is discussed further in section 4.2

Suivi~ and Antescofo

Antescofo is the most recent system of score following from IRCAM, headed by Arisha Cont. It is an external module for Max MSP (and Pd) and it makes use of a synchronous programming language to anticipate (hence the name) performance parameters as musicians do thus enabling computer realised temporal elements such as sound effects changing with the score of the piece.

The history of score following page at the IRCAM website shows the development of score following within IRCAM starting from the Vercoe system in 1983, through to Antescofo.⁵⁸ The notable other system from IRCAM is the Suivi~ using the probabilistic method of Hidden Markov Models (HMMs), a widely used system in score following and some beat tracking algorithms.

While Antescofo has polyphonic capabilities, the pieces that seem to suit the system best are where solo instruments are followed and the system interacts with them, for example pieces like the Pierre Boulez piece *Anthèmes 2*. The actual following of the score is precise and smooth with the computer following as a conductor would (the visualisation in the example given is in NoteAbility Pro).

Music Plus One

Music Plus One is a practice and performance aid developed by Christopher Raphael and is currently available as a standalone application for Windows. It takes its name from Music Minus One, the retailer of accompaniment CDs (that is, recordings of musical pieces missing an instrument to play along with). This system also makes use of HMMs and appears to follow just as precisely as Antescofo.

IAED (Intelligent Multi-Track Audio Editor)

As the name indicates, IAED is a tool for offline score following to enable audio correction of music in a similar way to many sequencers or Digital Audio Workstations (DAWs) but from a score-orientated paradigm. IAED is developed by Roger Dannenberg at Carnegie Mellon University, and is written in C++, but is not readily downloadable at this moment. This work is particularly noteworthy as Dannenberg is generally quoted in academia as one of the founders of beat tracking and score following along with Vercoe.

⁵⁸ The videos can be found on the IRCAM IMTR Score Following page: http://imtr.ircam.fr/imtr/Score_Following_History [accessed 15 August 2010].

4.1.2.2 Commercial Score Followers

While the commercial application of beat tracking has been limited thus far to InTime, whose lack of popularity is indicated by its lack of Windows Vista or Windows7 update, score following boasts three up-to-date and well-marketed products. While this follows the trend for more development in score following compared to beat tracking the main reason is the relatively abundant market for score following as practice aids, performance aids and, particularly, teaching tools, the one application which is used by both classical and popular music. Specifically, there are far more people learning an instrument than wanting to push the boundaries of popular music technology.

While none of the products mentioned here openly discuss their methods of score following, it seems reasonable to assume that some kind of flexible system based on probability is used, such as HMMs.

Score-Follower

Score-Follower is developed by Tonara and is marketed as a polyphonic score follower specifically for the use of automatic page-turning rather than automatic accompaniment. It accepts both audio and MIDI input and uses a standard MIDI format for the scores. The demonstrations of Score Follower show its use with a single microphone and several instruments being played and tracked simultaneously.

Home Concert Xtreme

Home Concert Xtreme is developed by TimeWarp Technologies and is notable for being patented under US patent 6166314 (2000).⁵⁹ The system is for piano keyboards and works using a MIDI input so it is a particularly limited product, but one that is likely to be convenient to use and fairly inexpensive at \$99.

⁵⁹ Frank M. Weinstock and George F. Litterst, 'Method and Apparatus for Real-Time Correlation of a Performance to a Musical Score', United States Patent 6166314, (2000).

Smart Music

Smart Music is developed by MakeMusic and is marketed as a learning environment. It allows automated distance learning through 'teachers' sending students customised assignments and automated classroom learning. In both cases students watch their performance in front of their computer to get instant feedback. Without getting into the pedagogical advantages and drawbacks of a system such as this, the system seems to track a wide range of instruments well using the basic audio input of a standard computer.

4.2 Why is Real-Time Rhythmic Automatic Accompaniment for Popular Music so Rare?

As mentioned in 4.1.1, there are very few studies in real-time rhythmic automatic accompaniment for popular music. The main reason for this is that popular music is not traditionally regarded as a field of academic study. While this rule is being tested and re-negotiated all across the Western world of music academia, it is an understandable, if sometimes mistaken, generalisation that classical/art music is structurally more complex and requires a higher level of performance skill. It is beyond the scope of this dissertation to consider the enormous range of reasons behind this highly debated musical divide. The effect on development of real-time rhythmic automatic accompaniment in popular music is such that it is left up to the performers themselves to create and implement these systems.

In the world of art/classical music there are resources to push the boundaries of what technology can bring to the art form so that interdisciplinary panels of technologists (i.e. IRCAM) work with internationally renowned composers (i.e. Pierre Boulez) and develop what we now see as well-established systems to further the art form.

Traditionally, most performers (either classical or popular) are not technically adept enough to create these systems. Most popular musicians, when contemplating a backing track simply turn to the studio version of the songs and take out the elements to be played live. Despite the occasional breaking down of the boundary between performing musician and technologist in the popular music world, there is still this

traditional divide that holds musicians back from learning the technical skills and exploring the creative opportunities within technology. What is apparent then, when developing and researching these systems, is that while performing musicians may like the idea of a flexible accompaniment system, the actual parameters of the specific systems to be used on stage are generally unclear.

While it is worth mentioning that the world of popular music does have an element of commercial input that can encourage an amount of musical change or at least use of interesting new musical creation tools (for example, The Prodigy's much touted use of the music production software Reason⁶⁰), this demands a far more measurable return on its investment, with far less freedom or incentive to allow the music to become more challenging and 'less like the CD'.

Thankfully, the divide between musician and technologist in the popular music world is slowly narrowing and with it performers and technologists such as Andrew Robertson and Nick Collins are working to make systems of real-time rhythmic automatic accompaniment a reality. Also, the stigma of popular music as a field of study is slowly lifting and unveiling some extremely interesting and challenging avenues of research.

That the concepts of score following and beat tracking had become rigid in their use does not help the development of real-time rhythmic automatic accompaniment systems for popular music. In the following system proposal there is much blurring of the boundaries and exploration of the areas in-between the standard methodologies, while attempting to remain focused.

Score following is not well suited to popular music as there is not easy access to the scores because they are not widely used in performance. Also, the score following paradigm is too rigid allowing no means of structural flexibility. Beat tracking works well for repetitive loops, but there is no means of those loops changing at the correct time. The musician would be forced to use a single repeating phrase and although this phrase would react to tempo changes this is a very limited use of technology. What is needed, then, is either a flexible version of score following (for popular outfits adept at scoring) or a beat tracker that can react to cues and change the accompaniment accordingly.

⁶⁰ Fredrik Hägglund, 'Liam Howlett of Prodigy Explains Why Reason, to him, is the Best Thing Since Sliced Bread', *Propellerheads Website*, (2003) <http://www.propellerheads.se/substance/artist-stories/index.cfm?fuseaction=get_article&article=liam-howlett> [accessed 19 August 2010].

4.3 Classification of Real-Time Automatic Accompaniment for Popular Music

As discussed in section 3.3, the current beat trackers and score followers are classified generically under MIR and evaluated in the MIREX framework. Although there is excellent work done through ISMIR and MIREX and it is certainly necessary to encourage large-scale evaluation, this also obscures the actual use of the algorithms as the evaluation is based on a figure of accuracy that can only be a compromise between the different outcomes optimum to the applications. For a discussion of the different applications of beat tracking see section 3.2.

There is also the issue that there is an inherent bias toward offline processing due to the massive demand for automatic indexing and categorisation in modern databases and libraries. Also, casual beat tracking algorithms are at a serious disadvantage if judged alongside non-causal beat tracking algorithms and will always appear inferior due to their nature. This may have been the reason for the Davies algorithm results being pulled from MIREX 2006.

Another, more balanced and accurate means of evaluation would be to separate real-time beat tracking and categorise it under HCI. This re-categorisation is implicit in the presentation of beat tracking algorithms at the conference for New Instruments for Musical Expression (NIME), a conference with many elements that could be considered a highly specialised area of HCI.

While systems of real-time rhythmic automatic accompaniment are not acting as direct keyboard and mouse replacements, it is clearly a method of interacting with a computer and certainly not too far removed from the tap to tempo function using the space bar and or mouse. This is essentially a highly simplified beat tracker and is found as a function of nearly every modern sequencer. There would not be any reason to use a causal beat tracker in an offline situation as this would be introducing unnecessary system restraints, so any overlap is highly unlikely. It would make sense to simply categorise real-time beat tracking under real-time rhythmic automatic accompaniment as a sub category of HCI. The obvious choice for an institution to manage this would be the Centre for Digital Music at Queen Mary, University of London as all of the effective real-time beat trackers discussed in this dissertation came from there.

It would, of course, be counter-productive to implement a score following evaluation too similar to the MIREX task, but if such score follower, developed for

popular music, were sufficiently different to the academic systems discussed in section 4.1.2.1, then it might be worth categorising under real-time rhythmic automatic accompaniment.

5. Suggested Implementation

5.1 System Overview

This proposed system of real-time rhythmic automatic accompaniment is non-generative, that is, the accompaniment is pre-programmed, but in a form flexible enough to be manipulated. Such a system requires two main elements: the control element and the playback element (see Figure 4). So far only the control element has been discussed as this is by far the most novel and challenging aspect. However, the playback element will be explored in section 5.2.6.

As discussed in section 4.2, the kind of approaches to controlling a system of real-time rhythmic automatic accompaniment that would be of specific use to popular musicians would be either a flexible score follower or a cue responsive beat tracker. A clear definition, the possible parameters and the outlines of a cue responsive beat tracker and a flexible score follower are discussed in sections 5.1.1 and 5.1.2, respectively.

5.1.1 Cue Responsive Beat Tracking

A system of cue responsive beat tracking is based on the way that a musician intuitively plays, improvises or jams with other musicians when the structure of the music is not fixed. As the vast majority of popular music is written in repetitive sections, the important thing for a musician to know is when to change to the next section. It is not possible to concentrate on all other elements of the music, while also concentrating on what the musician is playing, so the musician will listen for a cue that the next section is coming up, typically by a vocal cue as this is generally the lead part.

Of course, many experienced musicians in a familiar ensemble will be able to know where they are and will be able to 'feel' that a change is approaching, but this is based on a holistic system of learning and picking up on a multitude of nuances, something that would be far harder to emulate in software. It is also reassuring to these players that they can fall back on cues in case something unexpected happens.

For a system of cue responsive beat tracking, the cue would be more likely to be direct rather than musical. A musical cue could be from any instrument, but for popular music it is likely that the beat tracking will be driven by the drums, so the beat tracker, while also tracking the beats of the music will also be looking for a particular series of hits to tell it to change accompaniment. A direct cue would most likely be a MIDI trigger, activated by either a drum pad, or a footswitch, which would tell the system to change accompaniment at the end of that bar.

To use musical cues, the system is effectively employing a small element of score following, that is, instead of following the score note by note, it will be looking for key elements that form a cue, which would be accurately detected and not easily mistaken. To find the most effective cue types a lot of experimentation would need to take place, which could even cover ideas such as word recognition for lead vocals.

Initially the accompaniment will have a linear musical structure. For example: the accompaniment for section A could be followed by section B, then potentially return to section A and then go on to section C, but while playing there would be no alternative other than progressing onto the programmed section.

However, it would be possible to have a non-linear operation where the system would have different options, so after section A the system could either progress onto section B, or section C depending on the cue, and so forth. This could be designed into the system and depicted as a flow chart of choices.

5.1.2 Flexible Score Following

The proposed system of flexible score following takes the problem from the opposite perspective to that of the cue responsive beat tracker. It would be programmed to know and follow the score as a conventional score follower, but it would be possible to override or update the score in real-time at the whim of the lead musician. This would be particularly hard to achieve with a musical cue as the end of a section to be repeated would sound nearly identical to the end of that section to be followed by a different section. The cue would be the equivalent to the musical directions shouted onstage like the directions James Brown famously made a part of the live song 'Sex Machine': "Take 'em to the bridge? Hit me now!"⁶¹

⁶¹ James Brown, *Sex Machine*, King Records, (1970) [on LP].

It would be interesting to be able to direct a musical score by shouted directions or even through dancing as James Brown did, however this is somewhat beyond the scope of this dissertation. A more practical method would be a direct control, for example a foot switch sending a MIDI message that puts repeat markings at the end of that bar and could set for the number of bars most likely to be wanted.

As a customisation to an existing system, this would be a far more simple procedure than that of cue responsive beat tracking, as the hard part is the score following itself. There would need to be a contingency for when the score follower does not recognise the notes being played though, for example if there is a period of collective improvisation. In this case, the score follower would need to fall back on beat tracking to carry on the accompaniment.

It is interesting to note, however, that within each approach, elements of the other are employed. For example, an element of score following would be potentially employed in musical cues of the cue responsive beat tracker and the score follower might need to fall back on beat tracking. It is even possible in the longer term that they could potentially be two different modes of driving a more comprehensive system of real-time rhythmic automatic accompaniment.

Due to the lack of notation usage in popular music I do not believe that a system derived from score following would be as useful, nor as likely to work as well. Because of this I will be concentrating on the cue responsive beat tracking system.

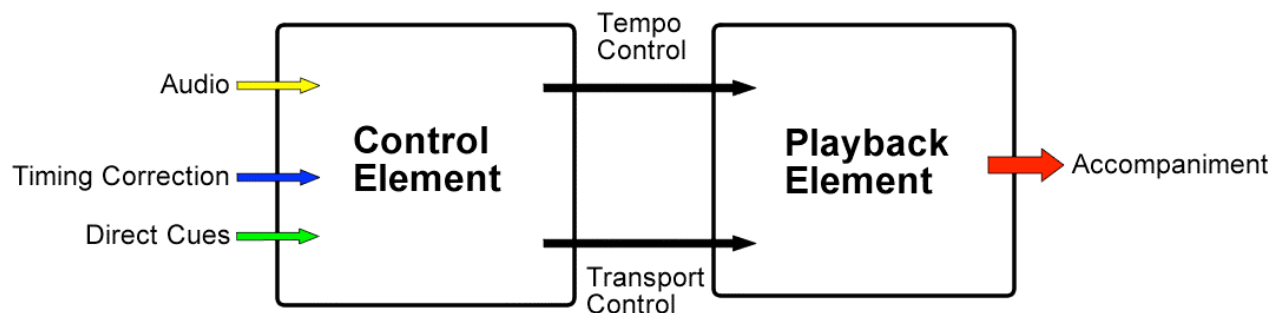
5.2 System Elements

5.2.1 System Outline

A basic outline of the proposed system is that of a control element driving a playback element, which is shown in Figure 4. The control element is the cue responsive beat tracker, which receives both audio and control messages (direct cues and any timing correction) and uses these to dictate the tempo and transport operations to the playback element. The playback element in this proposed system is a third party sequencer (or Digital Audio Workstations (DAWs) as they are often marketed). The main reason for using a third party sequencer is that there are several reliable and widely used examples of sequencers that can be used, allowing the development of the cue responsive beat tracker to be concentrated on. This choice is discussed and explained further in section 5.2.6).

The programming environment for the cue responsive beat tracker is Max MSP, specifically Max 5, because of its flexibility, its large user base and the excellent support.

Figure 4. Basic System Elements



5.2.2 Audio Inputs

For the purposes of system design it would make sense to stick with the standard digital audio of 44.1KHz 16 bit, as this also is the default for the object btrack~. Initially there

will be two versions of the real-time rhythmic automatic accompaniment, which, after testing, will be combined so that there are two modes of operation:

The first mode of operation would be for the use of driving the system with a drum kit, specifically a kick drum and a snare drum, much like B-Keeper. These are the drums most likely to convey the dominant elements of the beat and by standard convention track one would be kick drum and track two would be snare drum.

The second mode of operation would be for the use of driving the system by a singer songwriter, specifically a vocal and an acoustic guitar (ideally from a piezo or standard pickup in the guitar). While standard live desk convention would be to put the vocals at the top of the tracks, here for the sake of starting with the dominant part it would make sense for the vocals to be track one and the guitar to be track two. Because of this separation of signals, it will be necessary to test different beat trackers to see which is the most reliable.

5.2.3 Control Inputs

The control inputs will be the timing control input and the direct cue input, triggered by either footswitches, such as the Logidy UMI3 (a configurable three switch USB/MIDI Parametric Foot Controller) or by drum pads such as the Roland Octapad SPD30 (an eight pad USB/MIDI drum trigger/module/sampler). The MIDI messages will be managed by Max MSP to execute the following functions.

5.2.3.1 Timing Control

Count in

There will be a need to start the system at the intended time and tempo and while an initial tempo could be specified in Max MSP (this will be experimented with), it would be far more musical to use a 'count in' function where the first four beats are tapped directly and then the accompaniment starts on the one beat. This would be achieved with the timing control input, which would act as a count in for the first four presses of a piece, then would be a 'one' corrector.

'One' Corrector

An element of timing correction will be necessary because although the state of the art beat trackers can give solid and reliable performance, they still suffer from occasionally tracking on the 'and' of the measure (see section 4.1.1). To correct this, B-Keeper uses a nudge function controlled by an operator that pushes an incorrectly tracked tatum (i.e. tracking the 'and') onto the correct beat (i.e. move back two semi-quavers). This needs several parameters including tatum multiple and direction control, and would be nearly impossible for a performer to operate while concentrating on playing. An alternative to this would be a 'one' corrector. The operation of a 'one' corrector is that the performer hits the switch or pad on what they perceive as the first beat of the bar. Max MSP will then find the closest tatum and reassign that as the one beat, effectively acting as an automated nudge control. This will need to be experimented with to find the optimum error window and tatum sensitivity, so that the beat tracker does not skip to the '4ah' or the '1ee' (see section 2.3 '1 ee and ah') and start playing back incorrectly.

This indicates that Max MSP will need to track not only the beat, but also the measure, however this will be far simpler with an effective 'one' corrector and is a relatively simple process using the MAX MSP timing functions.

5.2.3.2 Direct Cues

Initially, all cues will be direct rather than musical (see section 5.1.1), as this is a known and reliable method of control and so that development can be taken one step at a time.

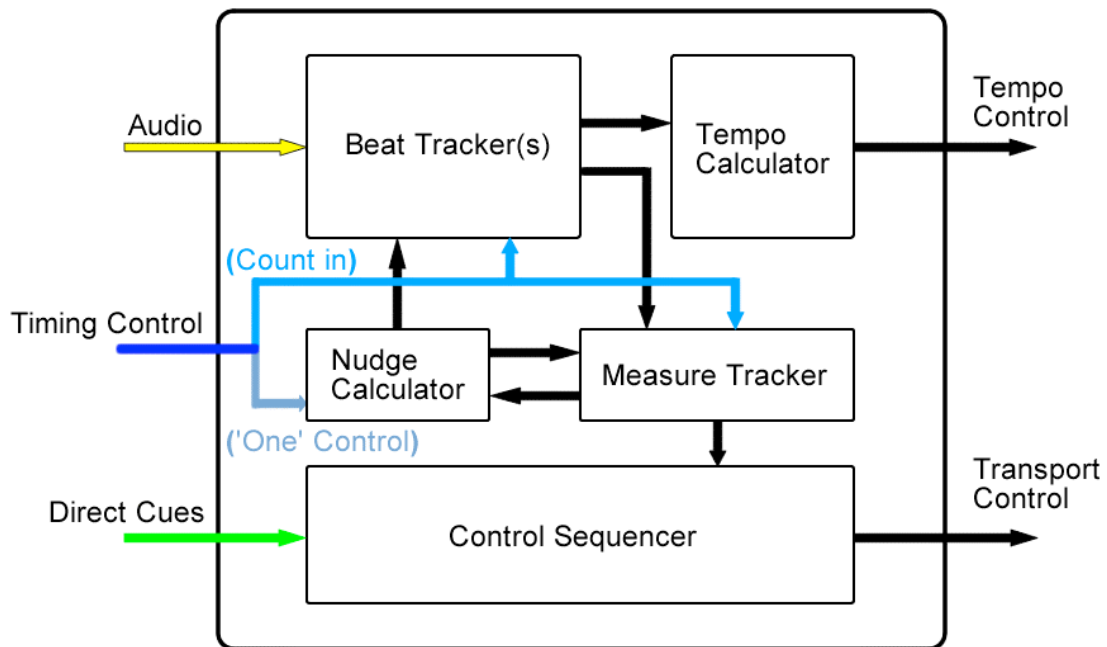
This also keeps computational load as low as possible, as musical cue recognition will need a significant amount of processing to be effective. (Computational load is discussed in section 5.2.7.) Effective use of cues will need to have two modes: a 'jump forward' and a 'repeat' mode. The 'jump forward' mode would be one quick press of the switch that will send the transport of the sequencer to the next cue at the end of the bar. The 'repeat' mode will be a continuously held press of the switch (for more than two seconds) that returns the transport to a pre-programmed loop point for as long as the switch is held, although it will only work when past the loop point.

Alternate Cues

Also, as discussed in section 5.1.1, initially musical structure of the accompaniment will be linear, but it would be a relatively simple matter to incorporate an alternative cue switch sending the sequencer to an alternative marker.

This considered, on the three switch UMI3, the timing correction would logically be the first switch, the primary cue would be the second switch and the alternative cue would be the third switch. A stop function could be easily programmed into either the footswitch if switch one is held down for an amount of time (or assigned to an out of way pad, or button on the Octopad). See Figure 5 for a visual representation of how the timing correction is integrated.

Figure 5. Overall Design of Control Element Components



Full Sequence Mode

The use of a sequencer (discussed in section 5.2.6) means that the standard means of accompaniment will be a pre-programmed full sequence mode, that is, each note from bar one to the end as opposed to a selection of loops. It is noteworthy that this use of a sequencer, driven by a beat tracker and counted in on a specific start point moves the system even closer to the realms of score following, as it knows whereabouts in each piece that it is. This original arrangement may prove to be used far more regularly than the flexible approach and the direct cues might hardly ever be used in a conventional live performance situation. Nevertheless, this would still be a valuable system in this scenario as there is an important distinction between being in control of an automated system or being controlled by it (i.e. following a backing track) and this system works toward a greater control of automated accompaniment in general.

Looped Mode

An interesting alternative to the standard sequenced accompaniment would be to program small loops that would repeat until the direct cue is pressed moving onto the next loop. This would be managed relatively simply by the control sequencer and would allow a mode of operation that would be particularly useful for more improvisational sessions, or even to use when composing with an ensemble. This mode is suggested here as an alternative to sequencing a full track as it is assumed that a fully sequenced piece will be more useful generally. This may not be the case of course and the looped mode does offer a method of use that is closer to the original concept of cue responsive beat tracking discussed in 5.1.1.

5.2.4 Beat Trackers

The actual beat trackers used in the control element will depend on quite extensive testing and involve three potential candidates: B-Keeper, btrack~ and Andrew Robertson's aubio objects. B-Keeper and btrack~ are both discussed in section 4.1.1.1. However, the aubio objects have not been included as they are not yet available. They come from audio, Paul Brossier's software library, and are developed by Andrew

Robertson. He kindly wrapped the driving function (an onset detector using complex spectral difference), into a Max MSP object and sent it to me for testing, but at the date of finishing this dissertation has not been able to send me the periodicity detector. For now, I am assuming that this beat tracker will be in the form of two objects, and I expect that they will be strong contenders to be used in the control element.

The three different beat trackers will form the core of the control element during testing (although isolated so that only one runs at a time, otherwise it is unlikely that there will be sufficient processing power) and it may be that there will be three different modes of operation if each beat tracker has different strengths. For instance, I expect the B-Keeper beat trackers to be best for drums (as they are specifically designed for drums) but the btrack~ to be best for the guitar and vocal.

Initial parameters for the beat trackers will include: initial tempo, tempo maximum and minimum and any threshold parameter needed for the driving function. Other parameters will be necessary during development, however it would be best to keep the operation as simple and as 'stage proof' as possible.

5.2.5 Control Sequencer

The control sequencer is a mirror of the cue points in the main sequencer and ensures that the control element knows the current position in the piece. This is necessary so that the control element knows which cue is next, without this there would need to be a different switch for each cue. The control sequencer will also manage the loop points for the 'repeat' function (see section 5.2.3.2) if using a full sequence. The overall design of control element components is shown in Figure 5.

5.2.6 Sequencer

The main options for a playback element in a live situation are to use either the basic playback of a digital audio file (for example .wav, .aif or .mp3), or a sequencer that can generate this music as and when desired. A digital audio file could potentially be time stretched to change the tempo in real-time, a technique used commonly in sequencers like Ableton Live. However, this is an unnecessarily restrictive means of realising a system of real-time rhythmic automatic accompaniment, especially

considering the wide use of sequencers and the general compatibility of this method of music production. The use of a sequencer, rather than coming up with a more dedicated solution to playback in a system of real-time rhythmic automatic accompaniment does mean that an already well-established approach to music production will be employed. This may have the potentially adverse effect of narrowing the possibilities of the system and discouraging more creative solutions, but this also gives an immediate wealth of reliable and effective working methods and established music creation tools. Also, inventing a new means of musical playback is not the aim of this dissertation or system.

As mentioned in section 5.2.1 the choice to use a third party sequencer rather than develop one in Max MSP is due to the extensive amount of development that the control element will need. While a dedicated sequencer could be developed in Max MSP specifically for this system, at this moment this seems an unnecessary reinvention of very well trodden technology. Also, the most recent sequencers are incredibly sophisticated and flexible, with many features that could prove useful in a live situation. It may be an avenue of future development to create the application as a single standalone piece of software as the absence of a separate interface between two different programs would most likely make the program easier to set up and more computationally efficient.

With these points and the overall objectives considered, using a third party sequencer is the most measured and practical approach and the best way to ensure that this project works towards its initial aims (see section 1.1).

The choice of sequencer is primarily decided by its ability to be controlled by the Max MSP control element. At the moment there are only two choices that offer the ability to be driven in a flexible manner, either Ableton Live or Reaper. The most obvious choice of sequencer for combination with Max MSP is Ableton Live, as the system could be integrated using Max for Live, allowing a more convenient control of the sequencer. Ableton Live is also the only commercial sequencer designed specifically for live use. On the other hand, the combination of Ableton Live 8 and Max for Live is substantially more expensive⁶² than Reaper, a more traditional DAW, intended mainly for use in the recording studio. The notable feature of Reaper is that while it is not open source, there is an enthusiastic community of developers working with a very flexible API (Application Programming Interface), which would mean with some coding, a Max Patch could drive

⁶² As of the 24 August 2010, Ableton Live 8 and Max for Live costs £285, (or £235 with an educational discount). This compares to Reaper 3, which costs £39.

the tempo and transport of Reaper as intended. Both applications will be tested to see which is best suited, but at this point it seems that the combination of Max5, Max for Live and Ableton Live will be the most effective.

5.2.7 Computational Load

A vitally important consideration when developing this system will be computational load. While computational load is always important, it is especially important here as the beat tracking algorithms are a very heavy load. For example, when tracking, btrack~ showed a standard usage of around 10–25 per cent but was peaking at 55 per cent. (The system is a 2.1 Ghz Inter Core 2 Duo Macbook running OSX 10.6.4 with 4G RAM). This means very little processing left for the sequencer, let alone any musical cue recognition should this element be pursued.

System optimisation will have to be an important part of the development, including the use of sequencer. Any effects will have to be processed beforehand rather than run as plugins, and virtual instruments will need to be made as efficient as possible. It may also be necessary to network several computers if there is a particularly demanding sequencer session.

6. Conclusion

The exploration of such a complex and interdisciplinary area as real-time rhythmic automatic accompaniment has proved to be a far greater challenge than I had expected, requiring a very large and varied set of sources. The categorisation and analysis of the factors involved has been particularly interesting and has proved a thorough grounding in the wider aspects of this project. Investigation into areas such as feed forward neural networks, experimental brain research and the history of synthetic performance, among others, has shown just how holistic an application such as this is.

It has been a fascinating discovery of all the people working on similar creative ideas, particularly as the modern use of the internet allows such global and instant democratisation of information, projects and ideas. It is notable just how many excellent valid and reliable sources have been found through the internet. In fact, the most useful sources have been electronic. The initial stages of system development would have been nearly impossible if not for the help and information found through forums (see Appendix B).

The real surprise of this project was the performance of the most recent beat trackers, which has been extremely encouraging for the prospects of a fully working system of real-time rhythmic automatic accompaniment as suggested in section 5.2. Prior to exploring B-Keeper and btrack~, there was a very real possibility that even the most up-to-date technology was just not sophisticated enough to be able to supply an accurate enough beat tracker for the system. In fact, the general impression that I have received from musicians, academics and laypeople alike was that I would quickly come down out of the science fiction fantasy idea when I started to look at the practical limitations of technology. While still not wanting to be unrealistic about the prospects of this real-time rhythmic automatic accompaniment system, it does look like there may be a real possibility of this being able to work on stage. The aims and objectives set out in the beginning of this academic year, in section 1.1, have been comprehensively met, although not without the occasional false lead, or conflicting term to wring out. To sum up, I am very much looking forward to the challenge of planning, developing and experimenting with this system.

Bibliography

- Chen, Ching-Wei and others, 'Improving Perceived Tempo Estimation by Statistical Modeling of Higher-Level Musical Descriptors', *Proceedings of the Audio Engineering Society Convention*, (2009)
- Bloch, Joshua J. and Roger B. Dannenberg, 'Real-time Computer Accompaniment of Keyboard Performances', *Proceedings of The International Computer Music Conference*, (1985), 279–89
- Bryson, Jonna, 'The Reactive Accompanist: Adaptation and Behaviour Decomposition in a Music System', *The Biology and Technology of Intelligent Autonomous Agents*, ed. by Luc Steels (Heidelberg: Springer-Verlag, 1995), pp. 365–76
- Byrd, Donald and Michael Fingerhut, 'The History of ISMIR – A Short Happy Tale', *D-Lib Magazine*, Vol. 8 No. 11, (2002)
<<http://www.dlib.org/dlib/november02/11inbrief.html#BYRD>> [accessed 9 August 2010]
- Collins, Nick, 'DrumTrack: Beat Induction from an Acoustic Drum Kit with Synchronised Scheduling', *Proceedings of The International Computer Music Conference*, (2005)
<<http://www.cus.cam.ac.uk/~nc272/papers/pdfs/drumtrack.pdf>> [accessed 10 August 2010]
- Dannenberg, Roger B., 'An On-Line Algorithm for Real-Time Accompaniment', *Proceedings of the International Computer Music Conference*, (1984), 193–98
<www.lb.cs.cmu.edu/~rbd/papers/icmc84accomp.pdf> [accessed 24 June 2010]
- Davies, Matthew E. P., 'Towards Automatic Accompaniment, (unpublished doctoral thesis, University of London, Queen Mary, 2007)
- Davies, Matthew E. P. and Mark D. Plumbley, 'Context-Dependent Beat Tracking of Musical Audio', *IEEE Transactions on Audio, Speech, and Language Processing*, 15:3 (2007), 1009–20
- Davies, Matthew E. P. and Paul M. Brossier and Mark D. Plumbley, 'Beat Tracking Towards Automatic Musical Accompaniment', *Proceedings of the Audio Engineering Society Convention*, (2005)

Davies, Matthew E. P., Norberto Degara and Mark D. Plumbley, 'Evaluation Methods for Musical Audio Beat Tracking Algorithms', *Technical Report C4DM-TR-09-06*, Queen Mary University of London, Centre for Digital Music, (2009)

<<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.152.6936&rep=rep1&type=pdf>> [accessed 31 March 2010]

Degara, Norberto and others, 'A Mixture-of-Experts Approach for Note Onset Detection', *Proceedings of the Audio Engineering Society Convention*, (2009)

Desain, Peter and Henkjan Honing, 'Tempo Curves Considered Harmful', *Contemporary Music Review*, 7 (1993), 123–38

Dixon, Simon, 'Automatic Extraction of Tempo and Beat from Expressive Performances', *Journal of New Musical Research*, 30:1 (2001), 39–58

Dixon, Simon, 'Evaluation of the Audio Beat Tracking System Beatroot', *Journal of New Musical Research*, 36:1 (2007), 39–50

Dixon, Simon and Werner Goebel and Emiliios Cambouropoulos, 'Perceptual Smoothness of Tempo in Expressively Performed Music', *Music Perception*, 23:3 (2006), 195–214

Downie, J. Stephen, 'Music Information Retrieval', in *Annual Review of Information Science and Technology* 37, ed. by Blaise Cronin, (Medford, NJ: Information Today, 2003) pp. 295–340

Downie, J. Stephen, 'The Music Information Retrieval Evaluation Exchange (2005–2007): A Window into Music Information Retrieval Research', *Acoustical Science and Technology*, 24:4 (2008), 247–55

Evans, Michael J., 'Interactive Beat Tracking for Assisted Annotation of Percussive Music', *Proceedings of the Audio Engineering Society Convention*, (2007)

Goto, Masataka, 'An Audio-Based Real-Time Beat Tracking System for Music With or Without Drum Sounds', *Journal of New Musical Research*, 30:2, (2001), 159–71

Goto, Masataka and Yoichi Muraoka, 'A Beat Tracking System for Acoustic Signals of Music', *Proceedings of the Second ACM International Conference on Multimedia*, (1994), 365–72

- Gouyan, Fabien and others, 'An Experimental Comparison of Audio Tempo Induction Algorithms', *IEEE Transactions on Audio, Speech, and Language Processing*, 15:3 (2006), 1832–44
- Gruhne, Matthias and others, 'An Evaluation of Pre-Processing Algorithms for Rhythmic Pattern Analysis', *Proceedings of the Audio Engineering Society Convention*, (2008)
- Hainsworth, Simon, 'Techniques for the Automated Analysis of Musical Audio', (unpublished doctoral thesis, University of Cambridge, 2003)
- Hägglund, Fredrik, 'Liam Howlett of Prodigy Explains why Reason, to him, is the Best Thing Since Sliced Bread', *Propellerheads Website*, (2003)
<http://www.propellerheads.se/substance/artist-stories/index.cfm?fuseaction=get_article&article=liam-howlett> [accessed 19 August 2010]
- Harrison, Gavin, 'Preparing Backing Tracks For Live Use', *Sound on Sound*, November 2003, <<http://www.soundonsound.com/sos/nov03/articles/backingtracks.htm>> [accessed 3 May 2010]
- Jehan, Tristan, 'Downbeat Prediction by Listening and Learning', IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, (2005)
<web.media.mit.edu/~tristan/Papers/WASPAA05_Tristan.pdf> [accessed 10 August 2010]
- Jordanous, Anna and Alan Smaill, 'Investigating the Role of Score Following in Automatic Musical Accompaniment', *Journal of New Musical Research*, 38:2 (2009), 197–209
- Longuet-Higgins, H. Christopher and Mark J. Steedman, 'On Interpreting Bach', in *Machine Intelligence 6*, ed. by Bernard Meltzer and Donald Michie (Edinburgh: Edinburgh University Press, 1971), pp. 221–41
- McKinney, Martin F. and Dirk Moelants, 'Ambiguity in Tempo Perception: "What Draws Listeners to Different Metrical Levels?"', *Music Perception*, 24:2 (2006), 155–66
- McKinney, Martin F. and others, 'Evaluation of Audio Beat Tracking and Music Tempo Extraction Algorithms', *Journal of New Musical Research*, 36:1 (2007), 1–16

- Meudic, Benoit, 'A Causal Algorithm for Beat-Tracking' *IRCAM*, (2002)
<<http://articles.ircam.fr/textes/Meudic02b/>> [accessed 21 January 2010]
- Parncutt, Richard, 'A Perceptual Model of Pulse Saliency and Metrical Accent in Musical Rhythms', *Music Perception*, 11:4 (1994), 409–64
- Pikrakis, Aggelos and Sergios Theodoridis, 'An Application of Empirical Mode Decomposition on Tempo Induction from Music Recordings', *Proceedings of the International Society for Music Information Retrieval Conference*, (2007)
- Raphael, Christopher, 'Automatic Segmentation of Acoustic Musical Signals Using Hidden Markov Models', *IEEE transactions on Pattern Analysis and Machine Intelligence*, 21:4 (1999), 360–70
- Raphael, Christopher, 'A Hybrid Graphical Model for Aligning Polyphonic Audio with Musical Scores', *Proceedings of the International Society of Music Information Retrieval*, (2004)
- Repp, Bruno H., 'Sensorimotor Synchronization: A Review of the Tapping Literature', *Psychonomic Bulletin & Review*, 12:6 (2005), 969–92
- Repp, Bruno H. and Peter E. Keller, 'Self Versus Other in Piano Performance: Detectability of Timing Perturbations Depends on Personal Playing Style', *Experimental Brain Research*, 202 (2010), 101–10
- Robertson, Andrew and Mark D. Plumbley, 'B-Keeper: A Beat-Tracker for Live Performance', *Proceedings of the New Interfaces for Musical Expression Conference*, (2007), 234–7 <www.elec.qmul.ac.uk/people/markp/2007/RobertsonPlumbley07-nime.pdf> [accessed 4 June 2010]
- Rossiter, David and Raymond Tang, 'Beat Deviation for Tempo Estimation Algorithms', *Journal of the Audio Engineering Society*, 55:11 (2007), 967–80
- Ryynänen, Matti P. and Anssi P. Klapuri, 'Automatic Transcription of Melody, Bass Line, and Chords in Polyphonic Music', *Computer Music Journal*, 32:3 (2008), 72–86
- Scaringella, Nicola and Giorgio Zoia, 'A Real-time Beat Trackers for Unrestricted Audio Signals', *IRCAM*, (2004)
<<http://recherche.ircam.fr/equipes/repmus/SMC04/scm04actes/P36.pdf>> [accessed 21 January 2010]

- Scheirer, Eric D., 'Tempo and Beat Analysis of Acoustical Musical Signals', *The Journal of the Acoustical Society of America*, 103:1 (1998), 588–601
- Simon, Ian, Dan Morris and Sumit Basu, 'MySong: Automatic Accompaniment Generation for Vocal Melodies', *Proceedings of the ACM Conference on Human Factors in Computing Systems*, (2008) <<http://research.microsoft.com/en-us/um/.../mysong/mysongchi2008.pdf>> [accessed 22 July 2010]
- Stark, Adam M., Mark D. Plumbley and Matthew E. P. Davies, 'Audio Effects for Real-Time Performance Using Beat Tracking', *Proceedings of the Audio Engineering Society Convention*, (2007)
- Stark, Adam M., Matthew E. P. Davies and Mark D. Plumbley, 'Real-Time Beat-Synchronous Analysis of Musical Audio', *Proceedings of the International Conference on Digital Audio Effects*, (2009) <dafx09.comopolimi.it/proceedings/papers/paper_65.pdf> [accessed 13 August 2010]
- Vauk, Ingo, 'Circular Login InTime', *Sound on Sound*, June 2005 <<http://www.soundonsound.com/sos/jun05/articles/intime.htm>> [accessed 15 August 2010]
- Vercoe, Barry, 'The Synthetic Performer in the Context of Live Performance', *Proceedings of the International Computer Music Conference*, (1984), 199–200 <www.media.mit.edu/~bv/papers/synthetic%20performer.pdf> [accessed 24 June 2010]
- Weinstock, Frank M. and George F. Litterst, 'Method and Apparatus for Real-Time Correlation of a Performance to a Musical Score', United States Patent 6166314, (2000)
- Xiao, Linxing, 'Using a Statistic Model to Capture the Association Between Timbre and Perceived Tempo', *Proceedings of the International Society for Music Information Retrieval*, (2008) <http://ismir2008.ismir.net/papers/ISMIR2008_231.pdf> [accessed 18 February 2010]

Discography

James Brown, *Sex Machine*, King Records, (1970) [on LP]

Appendix Forum Communication

B1. Cycling '74 > Max For Live Forum

Ableton Live + Live 4 Max for a system of automatic accompaniment?

Post URL: <<http://cycling74.com/forums/topic.php?id=27966>>

13 August 2010

[Tim Canfer](#)

I am currently researching systems of real-time rhythmic automatic accompaniment (i.e. no more damned fixed click tracks!), for a Masters dissertation and am dying to stop writing about people doing interesting musical things and start doing them myself... So far the best option for 'popular' music (as opposed to 'art' music, electro-acoustic ect), seems to be either using B-Keeper and Ableton as a starting point and then seeing if I can customise that and maybe build on it to generate systems driven by pitched instruments, (I know that would be really hard). I am aware of Circular Logic's InTime, it looks good but not really flexible enough.

As I only have Max5 at this point, my question is - would it be worth shelling out on Ableton Live + Live 4 Max, and would that be flexible enough to control not only the global tempo but other parameters such as timing and velocity offsets for different tracks / instruments? Clearly this is the Max4Live forum, but is there anything else out there that could do a similar job?

Also, if that is the case and it is worth it - can anyone tell me if it would be worth getting the full studio bundle?

Hope you can help, many thanks

Tim Canfer

<http://www.timcanfer.co.uk>

13 August 2010

[Venetian](#)

hi Tim,

It could be. I certainly think you might build an interesting system using it.

I've just made a version of B-Keeper that runs in MaxForLive. This directly controls the tempo using live.object. That does away with a lot of the control buttons on B-Keeper that make it confusing. However, I still like the old one myself and am even gigging tonight using that. (with Higamos Hogamos)

You could look at the zelastic object for Max - I've never used it yet - but it does the time-stretching apparently. We have various beatTrackers that would be of use to you - e.g. btrack~ by Adam Stark - that may be able to tell you the beats in the guitar signal. What exactly would you want to do? Synchronise with a rehearsed piece or have some kind of generative accompaniment?

regards

Andrew

13 August 2010

[Noob4Life](#)

"So far the best option for 'popular' music..."

there's no rule or 'best' as far as I've been able to see... 'popular' musicians use so many different things. some even also use SuperCollider:

<http://supercollider.sourceforge.net/>

some use reaper:

<http://www.reaper.fm/>

(seriously, take a GOOD look at reaper, if i didn't use Max/MSP so much, i would use reaper for everything)

some use renoise:

<http://www.renoise.com/>

and so many other things(with scripting capabilities, reaper can definitely do so much that m4l can, renoise is adding similar scripting capabilities to their environment, the price of renoise and reaper is also bound to make a mockery of everything else... at least, to me, their price makes everything else look like a greedy, immoral, businessman's joke)

max/msp alone, however(similar to using supercollider alone) is also a great option. you don't need ableton at all. max/msp has ITM and so much more for all your 'popular' and 'non-popular' needs, with even more diversity...

(in Ableton, often the sound(that crunchy digital/interpolated sound that ableton has, especially when you do any kind of synthesis) is just too distinct to get away from: you can hear an ableton-user a mile away, unless they use extensive VSTs or stick religiously to sampling only)

However, last but not least, you shouldn't listen to a noob like me ;)

Instead, check out this 9 part series on British synth pop. I don't think any of these people would've wanted to adhere to M4L, instead, they would've wanted to build their own... M4L will never be as DIY as using Max/MSP or SuperCollider or PD or CSound or programming your own VSTs/AUs in C/C++ alone(let alone programming from the ground up), consider going as deep as the rabbit hole might take you(in my humble opinion, that's not the way of M4L).

http://www.youtube.com/watch?v=WeVRYPjcVXg&feature=player_embedded

14 August 2010

[Tim Canfer](#)

Thanks for your replies - re your questions andrew

"What exactly would you want to do? Synchronise with a rehearsed piece or have some kind of generative accompaniment?"

Initially - the former (basically, flexible score following for popular music), but then ideally tie in some elements of generative accompaniment.

Re Noob4Life's points - I had not thought of using reaper - to be honest Max5 is as deep down the rabbit hole as I have been, I am not sure if I am up to the task of scripting, I had a brief crack at SuperCollider last year and the time it took to make a sine A440Hz kind of put me off.

The appeal of Ableton and Max4live is that I wouldn't have to reinvent the sequencer wheel and could get to making interesting music relatively quickly. The price is pretty hefty though, but the idea of doing it all (and making it solid) in Max is appealing if a bit daunting.

Is there some way driving Reaper with a Max beat tracker / score follower?

Thanks again

Tim

15 August 2010

[Tim Canfer](#)

I forgot to mention - the objects by Adam Stark are superb, btrack~ is surprisingly effective. Highly recommended, thanks Andrew.

15 August 2010

[Tim Canfer](#)

Re Reaper, I have started an interesting post on their forum re Max for Reaper, some interesting possibilities there... (Hope it is ok to link to from here:

<http://forum.cockos.com/showthread.php?p=566256&posted=1#post566256>)

18 August 2010

[Noob4Life](#)

nice! i'm glad others here are interested in Reaper. when they add OSC Max->Reaper and vice-versa will become a cinch(sp?). Renoise just added OSC, and I love scripting in LUA so i'm also excited about that(both are ridiculously affordable)... but for some, i understand the vertical tracker thing of Renoise can be limiting due to the adherence to quantization(even if you can easily overcome it with creative delays, quantization can still prove a counter-intuitive way of working)...

but anyways, didn't mean to imply that it's not possible in M4L, either. but there are many ways you can do it all in Max5. another way, besides some LCD-scripted sequencer linked to ITM via transport and similar objects, might be to take a look at incorporating this new JSUI-based timeline object:

<http://cycling74.com/forums/topic.php?id=27194>

20 August 2010

[Tim Canfer](#)

Renoise looks really interesting, but a bit old school and as you say, quantized for my purposes.

Re controlling Reaper or another 3rd party sequencer,

do you know if it is possible to do the following using MIDI or rewire? :

send an instruction that at the end of the bar, that sends the transport to a predetermined bar number and continues.

or would I need to create a program that can access the Reaper API?

Thanks again

Can a max patch control transport of ableton?

Post URL: <<http://cycling74.com/forums/topic.php?id=28074#post-140080>>

18 August 2010

[Tim Canfer](#)

I am considering purchasing Ableton and Max4live and this is one of the clinchers...

Specifically I would be looking to send to markers and looping various sections.

(my previous post must have been to the wrong forum,

<http://cycling74.com/forums/topic.php?id=28029#post-139067>)

Hope you can help, Thanks

Tim

18 August 2010

[Noob4Life](#)

couldn't find anything directly about that in the docs:
<http://cycling74.com/docs/max5/vignettes/intro/doclive.html>

good question...

if it were possible it would be here:

http://cycling74.com/docs/max5/refpages/m4l-ref/m4l_live_object_model.html

i might try the "Song" properties('current_song_time' etc. and it looks like looping is in there, too, so you might not do markers, but rather reset the loop and current positionings creatively...)

maybe you could get a demo and try it all out?

or hopefully someone with more experience with that particular technique will answer, as i'm a bit too lazy/busy right now to try it myself...

(it's hard staying 'Noob4Life' but i do my best ;-)

1 week ago

18 August 2010

[Tim Canfer](#)

Thanks Mr Noob4life - interestingly it may be better to use Reaper and somehow figure out how to send messages to the API.

Although where i start is another series of posts altogether, albeit on the cockos forum (great title!)

T-T-T-Tricky.

19 August 2010

[Noob4Life](#)

"T-T-T-Tricky. "

'It's tricky to rock a rhyme!'

and please feel free to just call me 'noob'(that goes 4 everyone!) :)

(or 'Mr.&Mrs. Noob' as I choose to remain 'unaffiliated' in a web environment ;)

20 August 2010

[xanadu](#)

Hello Tim,

You might be interested in this topic

<http://cycling74.com/forums/topic.php?id=25753> .

Kind regards,

Xanadu.

20 August 2010

[ShellUser](#)

To answer the original question: No, it seems you can't.

There is a transport object in Max which can be used to control the global transport. Yet when I try to control a rewire slave it you seemingly need to utilize the rewire~ object and send an int into its left inlet to start the (then) so called 'rewire transport'.

Fun thing is that the reference guide for transport

(<http://cycling74.com/docs/max5/refpages/max-ref/transport.html>) does imply this to be possible.

(afaik starting a master transport on a rewire master also starts the transports on the slaves, not so here).

Anyway; when using the examples there in M4L it does not start Live's transport. Making me assume right now that this isn't possible.

EDIT:

And as usual when I'm relaxing with a beer I'm talking before testing..
Ok: the real answer to the original question is: "yes you can", sorry for the confusion.
With the "is_playing" property of the Song class you can check the state of Live's transport ('observable') and trigger it ('set').

Also see: http://cycling74.com/docs/max5/refpages/m4l-ref/m4l_live_object_model.html#Song (check the property list).

As for me: I need some sleep that that I start talking some common sense again :P

24 August 2010

[Venetian](#)

Hi Tim,

You can get it to start playing and you can trigger any specific loops you want. So given a quantisation, if say you have a beat tracker in max doing a pulse, a message could trigger your loop on the next pulse point at the required tempo.

What I've not seen is any control over Live's warp markers so although this may be possible(?) I doubt it is at present, but neither would it be so useful to you.

Re beat tracking - I've done classes of the comb filter beat tracker and will post up to you later this week after porting into a max object.

In order to control Live, you use a live object, observe a property and then send a message. The one to fire clips is weird, you need to send "set fire" message not just "fire"

try looking in Live->Library->Lessons->clipOperations.amxd if you get hold of a copy.

Andrew

30 August 2010

[Tim Canfer](#)

Wow - thanks for all the help. This is way over my head at the moment as I have yet to get my hands dirty with MAX4Live, but I'll hand the dissertation in and then I am free to crack on. Will report in as soon as possible with a little more than confused thanks...

Thanks again

B2. Reaper Forums > Reaper General Discussion Forum

Max for Reaper?

Post URL: <<http://forum.cockos.com/showthread.php?p=566225#post566225>>

15 August 2010

[TimCanfer](#)

I am brand new to reaper and I am working towards producing a system of automatic accompaniment, essentially a track backing track that will respond to tempo changes. I am a Max MSP user and do not really fancy recreating the sequencer, I am looking a creating a beat tracker / score follower in Max that can drive various parameters of a sequenced backing track (and even potentially generate some backing on the fly, but that would be later...)

The obvious choice, is Max for Live, but I really like the look, lack of prohibitive cost and potential for scripting in reaper, so my question is, can I drive the tempo and transport of

a reaper session using a Max Patch?

If so, would it be possible to also manipulate more subtle timing of different midi tracks (effectively a randomization factor and timing bias), allowing a more realistic ensemble?

Hope you can help.

Thanks in advance

Tim

[nofish](#)

Hi,

can't help with your questions, sorry.

Just wanted to suggest that there's also a dedicated developer forum here where you could maybe get more answers (in case you haven't seen it).

<http://forum.cockos.com/forumdisplay.php?f=35>

btw, there's a (to me as a programming noob) similar thread there:

<http://forum.cockos.com/showthread.php?t=63381>

[schwa](#)

don't know specifically how Max works, but there are various ways for an external application to control REAPER, including sending MIDI, sending windows-level messages, or talking to a custom REAPER extension (which would have full access to the REAPER API).

We are interested in adding OSC support to REAPER, but it doesn't exist yet.

[TimCanfer](#)

thanks for the replies, that is really interesting - actually I probably should have posted this on the developer forum, apologies.

It looks like using MIDI would be the easiest way initially - similar to what Andrew Robertson has done with B-Keeper and Ableton.

I am not a Max master by any means but, it is basically a object orientated programing language, boxes and patch cables essentially. It is tricky stuff for someone with no programing experience, but an excellent place to start for making interesting things, I make a synthesis tutorial to use in my classes with it -

(www.timcanfer.co.uk/tcsynthtutorial) Pd is the free version, just as good but not as well supported IMO.

As far as I understand it is pretty nifty with OSC though, but MIDI is solid so it looks like as good a place to start as any.

B3. Reaper Forums > Reaper Development Forum

noob control question...

[TimCanfer](#)

I am a complete developer noob, but I'm enormously impressed interested by this community and the mysterious world of developing.

I have a query about how much I can control reaper using an outside program, specifically Max5 (an object oriented, high level programming environment, but as low level as I have got so far...)

In short, can I control tempo and transport functions using MIDI messages, or would I need to create a program that can access the Reaper API?

Specifically, what I am hoping to do is mentioned at the end of this post:

<http://forum.cockos.com/showthread.p...225#post566225>

Many thanks for your time

Tim

[bradleyfilms](#)

There are 30 markers pre-built for you to use, plus transport control and tempo coarse/fine adjustment, all available from MIDI.

[TimCanfer](#)

Superb, thankyou.

Now all I need to do is to work out how to get a Max patch to send those messages to the API. I'll get the SDK and get playing.

(I assume that this is not possible with rewire?)