Interactive Real-time Musical Systems

Thesis submitted in partial fulfilment
of the requirements of the University of London
for the Degree of Doctor of Philosophy

Andrew Robertson

October 2009

Centre for Digital Music,
School of Electronic Engineering and Computer Science,
Queen Mary University of London

I certify that this thesis, and the research to which it refers, are the product of my own work, and that any ideas or quotations from the work of other people, published or otherwise, are fully acknowledged in accordance with the standard referencing practices of the discipline. I acknowledge the helpful guidance and support of my supervisor, Dr Mark Plumbley.

Abstract

This thesis focuses on the development of automatic accompaniment systems. We investigate previous systems and look at a range of approaches that have been attempted for the problem of beat tracking. Most beat trackers are intended for the purposes of music information retrieval where a 'black box' approach is tested on a wide variety of music genres. We highlight some of the difficulties facing offline beat trackers and design a new approach for the problem of real-time drum tracking, developing a system, B-Keeper, which makes reasonable assumptions on the nature of the signal and is provided with useful prior knowledge.

Having developed the system with offline studio recordings, we look to test the system with human players. Existing offline evaluation methods seem less suitable for a performance system, since we also wish to evaluate the interaction between musician and machine. Although statistical data may reveal quantifiable measurements of the system's predictions and behaviour, we also want to test how well it functions within the context of a live performance. To do so, we devise an evaluation strategy to contrast a machine-controlled accompaniment with one controlled by a human.

We also present recent work on a real-time multiple pitch tracking, which is then extended to provide automatic accompaniment for harmonic instruments such as guitar. By aligning salient notes in the output from a dual pitch tracking process, we make changes to the tempo of the accompaniment in order to align it with a live stream. By demonstrating the system's ability to align offline tracks, we can show that under restricted initial conditions, the algorithm works well as an alignment tool.

Acknowledgments

"They say it is a wise man who climbs Mount Fuji, but a fool who climbs it twice."

I would like to express my extreme gratitude to my supervisor Prof. Mark Plumbley for his constant support and encouragement during the creation of this thesis.

I'd like to take this opportunity to thank Dr. Nick Bryan-Kinns, Prof. Pat Healey, Dr. Simon Dixon, Dr. Josh Reiss and Prof. Mark Sandler for their advice and support. Thanks to everyone at the Centre for Digital Music for all your help. I look forward to carrying on working with you all: Kurt, Katy, Dan, Steve, Enrique, Andrew, Rob, Ruohua, George, Becky, Mike, Matthias, Martin, Samer, Amélie, Chris C, Chris H, Dan, Tim, Xue, Yves, Maria, Ivan, Beiming and Jean-Baptiste.

I would also like to thank all the musicians who have taken part in this research: James Sedwards, Jem Doulton, Al Pickard, Hook and the Twin, Rocco Webb, Nigel Hoyle, J. C. Caddy, Marcus Pearce, Adam Betts, Mark Heaney, Matt Ingram, Hugo Wilkinson, Joe Yoshida, Gregg Hadley and Tom Oldfield. Thanks also to Ned and Andrew Birkin, Carnival of Souls, Harry Escott, Alex Ward, Amber Marks and Billy Sherrer. Special thanks to Adam Stark for his willingness to man the camera and act as observer in the 'Turing Test', to Matthew Davies for his many insights into the complex problem of beat tracking and to David Nock for his support and drum skills. Thank you to the EPSRC for funding this work.

Lastly, I'd like to thank to my brother James 'Jimmy' Robertson and my parents for their enduring support. We did climb Mount Fuji, on my twenty-first birthday - overnight and in a gale force wind. Somehow a bottle of champagne managed to get up there too. The sunrise was incredible, but I doubt I'll manage it again. Maybe one day.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This thesis is concerned with the creation of interactive musical systems. In particular, we want to interpret a performance in real-time in order to develop automated systems that can engage with human players. This is a difficult task, requiring that we reliably update estimates for the tempo, phase and bar position. This would provide a framework to enable machine interaction in a performance by processing rhythmic and harmonic information with respect to the musical meter. Without such a framework for *understanding* the audio input, any response generated by a computer may not exhibit the musical sensitivity and synchrony that we expect from intelligent musicians. This is particularly true in genres such as rock and pop music, where the definition of a regular beat is vital and close synchronisation is required.

We will first look at a drum tracking system that can synchronise audio with a live drummer and investigate methodologies for evaluating such a system. Since there is a two-way interaction between the musician and the system's response, we will investigate ways to evaluate this in the appropriate context. We will then look at a technique for real-time multi-pitch detection in order that harmonic information from other instruments can be processed by the system.

Throughout this thesis, U.S. terminology for note durations will be used. Thus, crotchets are referred to as quarter-notes and quavers as eighth-notes.

1.1 Objectives and Motivation

> "The rock musician represents a type of musician for whom creative involvement with technology, with amplifiers, microphones, special-effect machines and computer-controlled synthesizers has become increasingly characteristic."
> Peter Wicke, *Rock Music: Culture Aesthetics and Sociology* [Wic90]

In the studio, musicians are increasingly reliant upon the computer as a means for recording and mixing. In Rhythm and Noise [Gra96], Theodore Gracyk puts forward the argument that rock music differs from other genres, such as classical and jazz, in that the audio recording is the actual aesthetic object which we refer to when we say, for example, the Rolling Stones' 'Satisfaction' or Led Zeppelin's 'Stairway to Heaven'. Live performances might seek to recreate the original recording or, in the case of Led Zeppelin, expand upon it in a semi-improvised manner, but the song remains defined by the recorded performance. This differs from classical music, where the aesthetic object is the score that is 'interpreted' live, with concerts regarded as instantiations of the piece. Prior to rock music, recordings attempted to capture or replicate the audio sensations of being present at a live performance. With rock music, musicians and producers used the studio to create sounds that place the listener in an *idealised* (and potentially physically impossible) location. Musicians are at liberty to overdub tracks and 'double-track' vocals to create a surreal effect of them duetting with themselves.

As an example, Paul Simon has described the recording process involved for "The Boxer": "It was recorded all over the place - the basic tracks in Nashville, the end voices in New York St Paul's church, the strings in New York Columbia Studios and voices there too."[Lei73] In conversation with Daniel Levitin [Lev97], Paul Simon recalls the method by which they achieved the cannon-like snare sound: "... the snare drum on "The Boxer" ...was recorded in the elevator shaft of the CBS studios in New York at 52nd Street. That was a pure Roy [Halee - audio engineer] sound. He situated the drum in the elevator shaft and he hit it and he

recorded that. It was just huge."

In addition, early experimentation in the sixties led to the discovery of effects such as flanging, phasing, delay, chorusing and distortion; all of which help to create the psychedelic collages present in many works from the time such as 'Sergeant Pepper's Lonely Hearts' Club' and 'Pet Sounds'. Gradually, as listeners, we have been accustomed to the sound of these recordings, with the result that musicians face a significant challenge when trying to replicate or compete with them live. Indeed, technological advances are intricately linked with the sounds created on record.

At present, if musicians wish to incorporate electronic parts or samples into a live performance, they often resort to having the drummer wear headphones and play in time with a click track. Whilst this is successful in so far as the electronic component is introduced at the correct time, the performers are forced to make several concessions. The use of a click track forces the piece to be at played a set (often uniform) tempo and so reduces the musical expression of the performance. The drummer who wears headphones is acoustically isolated from the performance. Other methods involve using some form of trigger to cue the part, however this is potentially inaccurate if it is cued by hand and this necessitates that the computer's 'performance' must be at some pre-determined tempo, hence forcing the musicians to synchronise to the accompaniment rather than vice-versa.

One major objective of this thesis is the creation of a system for rock music so that pre-recorded audio can be synchronised with a band without such concessions. In order to do so, we focus upon interpretation of the drum signal to design a system that reliably follows subtle changes in tempo. This research could lead to new possibilities for live performance that have only previously been possible within the studio environment. By designing systems which have the ability to listen and respond accordingly, exciting new possibilities for musical expression are created. Computers are exceedingly fast and precise at scheduling events. These might be musical events such as electronic parts, the automation of audio effects or the control of visual projections and lighting. Raphael [Rap04] describes the potential of computer technology to transform performance:

> "A welcome bonus of synthesizing the accompaniment with electronic instruments is the virtually unlimited technical capacity one inherits. In this way, compositions with nearly arbitrarily fast notes, arbitrarily complex rhythms, and superhuman complexity of interaction between live and synthetic performer are now possible."

If the computer can relate a performance to its higher level abstract representation, then its powerful time scheduling and ability to manipulate audio at a very low level, as seen in granular synthesis and low-level audio sequencing, could be incorporated into live performance in a truly responsive, dynamic manner. Similarly, Robert Rowe [Row01] has described how his personal motivation is due to the possibilities for the new compositional domains afforded by the development of machine musicianship. It is our experimental nature which seeks to extend thought-based symbolic processes into computer systems that can realise new conceptual forms as auditory phenomena.

1.2 Outline of Thesis

In order to build interactive systems, we require the analysis of the musical content of a live performance in real-time. We will examine two different approaches to the problem of accompaniment systems. One which makes use of rhythmic information and the other using pitch information. For rock bands, when drums are present, they provide a clear determination of the beat and our first approach to the problem is a system for drum tracking of music with a regular tempo. Where drums play a less defined role, harmonic information from other instruments may be required to synchronise with the performance and, in addition, interactive systems may require harmonic information, such as pitched notes, chords or key, for the generation of their musical response. With these purposes in mind, we present an algorithm for real-time multi pitch analysis that provides a MIDI representation of audio input.

Chapter Two
We present relevant background for our development of a real-time drum tracking system. We examine work in music psychology, beat tracking, the nature of drum signals and look at existing interactive systems and the programming languages and environments used to implement them.

Chapter Three
We describe the development of a novel system for real-time drum tracking.

Chapter Four
We evaluate the drum tracker using offline testing and a musical 'Turing Test', inspired by the famous test for machine intelligence proposed by Alan Turing.

Chapter Five
We present further modifications made to the drum tracking algorithm in light of the testing carried out in Chapter Four.

Chapter Six
We present work on a real-time multi-pitch tracking technique which estimates the amplitude of partials for each fundamental. These estimates are then used in detecting the fundamental and subtracted from higher frequencies to prevent false detection of notes. The pitch tracker is evaluated on offline piano music and contrasted to an online algorithm.

Chapter Seven
We present a conclusion on the work presented in this thesis and give an outline of possible directions for future work.

Appendix
Preliminary work is presented that makes use of the audio-to-MIDI algorithm described in Chapter Six as the input to a tracking algorithm

which aims to synchronise two audio sources from their MIDI representation. The tracking algorithm is evaluated using performances of Bach's 'Well-Tempered Clavier' by two different pianists.

1.3 Related Publications

D. Stowell, A. Robertson, N. Bryan-Kinns, and M. D. Plumbley. "Evaluation of live human-computer music-making: quantitative and qualitative approaches." *International Journal of Human-Computer Studies, Volume 67, Issue 11*, pages 960-975, November 2009.

A. Robertson and M. D. Plumbley, "Post-processing fiddle∼ : A real-time multi-pitch tracking technique using harmonic partial subtraction for use in live performance systems", in *Proceedings of the International Computer Music Conference*, pages 227-230, Montreal, Canada, 2009.

A. Robertson, M. D. Plumbley and N. Bryan-Kinns, "A Turing Test for B-Keeper: Evaluating an interactive real-time beat tracker", in *Proceedings of the 8th International Conference on New Interfaces for Musical Expression*, pages 319-324, Genova, Italy, 2008.

J.-B. Thiebaut, S. Abdallah, A. Robertson, N. Bryan-Kinns and M. D. Plumbley, "Real time gesture learning and recognition: Towards automatic categorization", in *Proceedings of the 8th International Conference on New Interfaces for Musical Expression*, pages 215-217, Genova, Italy 2008.

A. Robertson and M. D. Plumbley, "B-Keeper: A Beat-Tracker for Live Performance", in *Proceedings of the 7th International Conference on New Interfaces for Musical Expression*, pages 234-237, New York, USA, 2007.

A. Robertson and M. D. Plumbley, "Real-Time Beat Tracker for Live Performance with Drums.", in *Proceedings of the Digital Music Research Network Summer Conference*, Leeds Metropolitan University, UK, 7-8 July 2007.

A. Robertson and M. D. Plumbley, "Real-time Interactive Musical Systems: An Overview", in *Proceedings of the Digital Music Research Network Doctoral Researchers Conference*, London, UK, July 22-23, 2006.

1.4 Thesis Contributions

The principal contributions of this thesis are:

- A real-time beat tracking system designed for drum signals.
- A methodology of evaluation of this system based upon the 'Turing Test'.
- A real-time multi-pitch tracking technique for audio to MIDI conversion.

Chapter 2

Background

The major aim of this thesis is the construction of interactive accompaniment systems suited to rock and pop music. In these genres, although there is often no symbolic score, humans experience a strong perception of musical structure through regularity of beat and the timing of harmonic events. The role of the drums is central to this perception of regular rhythm and so we will look to build the foundations of our accompaniment system by using drum signals alone. Before attempting to construct a real-time system, we will examine the literature from a wide range of relevant areas including music psychology, drum signals, beat tracking and interactive systems.

2.1 Interactive Music Systems

We are interested in building an interactive accompaniment system and so we need to ask the question: "what defines an interactive system?" The definition for human-computer interaction proposed by the ACM states that "Human-computer interaction is a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them." Since this requires an intuitive understanding of what constitutes an 'interactive' system, this does not help us to specify the characteristics which might define such a system. In the literature we find conflicting views on what constitutes true interactivity.

Robert Rowe [Row93] characterises interactive computer music systems as "those whose behaviour changes in response to musical input."

This definition encompasses a wide variety of applications for computers within music that includes score following, automatic accompaniment, generative systems, improvisation systems and sound processing. Rowe distinguishes three processes are present within an interactive system: *sensing*, *processing* and *response*. Rowe presents a further categorisation of interactive systems. *Score-driven* and *performance-driven* systems can be distinguished by the extent to which stored musical fragments or traditional categories of organisation such as tempo and meter are used by the system to interpret the input. A performance-driven system does not have a stored representation of musical content as is available to a score-driven system, but might operate on the basis of other quantities such as note density or regularity. A system might also display a combination of these attributes. Another distinction can be made between the response methods as *transformative*, *generative* or *sequenced*. Transformative systems take existent musical material and apply transformations to it. Generative algorithms create musical output from more elementary source material, whilst sequenced techniques use pre-recorded fragments in response to input, making variations in tempo or dynamic shape. Finally Rowe distinguishes between *instrument* and *player* paradigms, depending on whether the system is considered as an augmented musical instrument or a separate player with a musical personality of its own.

Chadabe proposed the term 'interactive composing' to refer to the activity of using performable, real-time computer music systems to perform and compose music. The environment thereby created is one in which "the computer responds to the performer and the performer reacts to the computer, and the music takes it's form through that mutually influential, interactive relationship." Chadabe relates how such a system "operates as an intelligent instrument - intelligent in the sense that it responds to the performer in a complex, not entirely predictable way, adding information to what a performer specifies and providing cues to the performer for further actions." Random-number generators can be used to introduce a level of complexity to the computer's response and thereby achieve this element of unpredictability by the performer. However, he leaves open the possibility for future alternative complex generators, such as an artificially

intelligent, knowledge-based system, that may be used in the response algorithm.

Dobrian [Dob04] characterises interaction as 'the mutual influence between agents that are in some way autonomous decision makers'. Whilst the concept of mutual influence is useful in this context, Dobrian's conception of interaction is intended to apply generative systems and is motivated by the paradigm of improvisation. For Dobrian, in order for the system to be more than merely 'reactive', the computer must make decisions not fully predicted by the algorithm. This implies the machine's behaviour must include an element of chance, requiring the use of pseudo-random numbers to generate response in a stochastic fashion. Similarly, Dobrian does not regard a system as interactive if the musician plays from a score, since he regards the part as pre-determined. Whilst it is true that the abstract (or quantified) representation of the performance would be thereby predetermined, the musician is still free to *interpret* the score. This includes the dimensions of timbre, volume, tempo and timing. This illustrates a limitation of Dobrian's conception of what constitutes an 'interactive' system. He requires the element of interactivity to be located in the methods used to generate the abstract representation of the computer's output, but the complexity of the requirements involved in the process of playing this part in a musical manner are not considered. Given that two human musicians are considered to interact when playing scored parts, Dobrian's formulation is too strict to apply to the case of automatic accompaniment.

Although the recommendation for randomness might be useful as a means to creating a perceived autonomy on the part of the computer, in the context of a description of an interactive system it becomes proscriptive and appeals to structural properties of the algorithm rather than some musical or perceptual feature or the aesthetic result. Under a strict interpretation of this definition, interactive and reactive systems could not be distinguished by virtue of their behaviour but only by examining source code. It follows from Dobrian's argument, that it is simple to construct an "interactive" system from a merely "reactive" one by subsequently incorporating some quantity of pseudo-randomness in the algorithm. Yet it

may also be possible to incorporate pseudo-randomness in a deterministic algorithm by calculating functions as the result of multiple variables, so that the space is never sufficiently known to the performer to enable them to predict its response to a given input. A similar line of argument is found in Jordà's thesis [Jor05], where score following systems are discounted as intelligent but not interactive since the player is following "a predetermined path". Given the possibilities for expression by variation of timing, dynamics and timbre, it would appear that this path is only predefined in terms of pitch and quantised duration. Jordá states that "interactive music systems must be interactive enough to affect and modify the performer(s) actions, thus provoking an ongoing dialogue between the performer(s) and the computer system." However, since auditory feedback from a reactive accompaniment affects and modifies a performer's actions when the accompanist is human, then, in principle, the same influence should be possible from a computer-based system.

It is instructive here to look at some of the literature on musical improvisation. Benson [Ben03] regards many musical compositional processes as being essentially *improvisational* in nature. Whereas it is now common to separate the composer from the performer and to locate the musical work outside of any individual performance, Benson argues that improvisation has played an important role in shaping the work and that it is important to acknowledge the historical impact of prior performances in defining a musical piece. He traces the history of musical performance practice from its Renaissance and Baroque origins, where performers were expected to improvise and much of the music was sketched rather than fully realised. Benson holds the view that "composition and performance are improvisatory in nature, albeit in different ways and to differing degrees." Benson provides examples of different levels of improvisation that can be found in musical processes, from filling in details, such as tempo, timbre, attack and dynamics that are not explicitly stated in the score, adding trill notes and filling out chords, improvising Classical cadenzas, making arrangements of the score, changing the melody line or chords, to using the basic form of the score, such as a twelve-bar blues, to improvise

within its confines. Beyond these instances of performer-related improvisation, composers may improvise by using particular forms as a template, using particular pieces as the basis for a composition or working within a musical tradition, such as classical, blues or jazz, where the tradition itself is improvised upon. Thus, while the score suggests pre-determination, improvisation on the part of the performer is always required to some degree.

Berliner's [Ber94] comprehensive study of the practices of jazz musicians reveals insights on how they go about creating solo improvisations. Performance may consist of adding embellishments, such as grace notes, and variation of phrasing with subtle anticipations, or transformations where the original melody is still recognised. Berliner finds that "typically, ... players restrain themselves during the melody's formal presentation, reserving their most compositional activity for improvised solos." However, each musician's experimentation is guided by their knowledge of the jazz tradition and by music theory. Players may differ in their choice of notes relative to the underlying chords ('the changes'); some prefer the 'vertical' concept of articulating each chord, whilst others may favour a 'horizontal' concept of playing a phrase across the chord changes. In addition, players may be characterised by their tendency to choose notes inside or outside the chord tones. The theoretical concept of scales provides an alternative way of thinking about pitch relationships, where the scale can be conceived of as a combination of pitches both inside and outside of the chord. Musicians also build up a personal vocabulary of phrases that can be used when soloing. Thus, the act of improvisation is not to create music out of nothing without restrictions on structure and form, but to engage in a practice of inventive composing during performance.

In relation to the the previous argument, rather than requiring that interactive systems incorporate randomness, we might first aim to construct a system capable of analysing the underlying musical structures. The motivation behind Dobrian and Jordá's definitions emerge from Chadabe's conception of interactive composing and the requirement that the system not be entirely *predictable*. Should we be able to construct a real-time

system that can analyse structure, we could then create interactive systems that use this information to transform rhythm and melody using rules derived from music theories. The use of randomness might be used, as Chadabe suggests, to generate complexity within the response, but in the context of human-computer performances, this will be more effective when used in conjunction within the constraints of music theory and similar results might also be achieved through the use of interesting mapping strategies.

In this thesis, we shall define *interaction* as a "mutual influence between performer and system". This definition is in accord with that proposed by most commentators in the field and excludes particularities concerning interaction that might apply to specific fields such as generative systems. The common features in the design of such systems is that they accept audio or sensor-based input from a performer, process this according to system settings or parameters and output a musical or visual response. The interaction between musician and system thereby creates a feedback loop of mutual influence.

We are aiming to build a system that is a capable of following a performance and intuiting the musical structure that underlies it, thereby enabling other types of responsive system to be constructed that make use of the rhythmic and harmonic information. In order to do so, we shall examine work on previous accompaniment systems and other systems for interactive improvisation.

2.1.1 Score Followers

It is over twenty-five years since Barry Vercoe [Ver84] and Roger Dannenberg [Dan84] first independently presented work on the task of score following at the 1984 International Computer Music Conference.Dannenberg [Dan84] formalised the problem of score following as finding the longest common subsequence of two strings, one representing the score and the other the observed performance. Where notes are skipped, they must be removed from the score string. Where they are inserted, they are removed from the performance string and where they are wrongly played

or detected, they must be removed from both. His real-time accompanist employs dynamic programming to calculate the least cost match between the observed performance, translated into MIDI notes, and the score representation.

Vercoe's [Ver84] work on score following, 'Synthetic Performer', utilised both audio-to-MIDI conversion and optical sensors on the keys of the flute to follow the soloist through the score and provide an automatic accompaniment. Despite its ability to follow a wide variety of interpretations and negotiate errors and distortions of the score, it had no performance memory. Working in conjunction with Miller Puckette, the subsequent year he presented 'Synthetic Rehearsal' [VP85] which incorporates a mechanism to learn from previous renditions of the piece. Vercoe formulates the problem of score following as consisting of three processes: *Listen*, *Perform* and *Learn*.

Listen encompasses all potential input to the system, both audio and visual or mechanical information, such as provided by his use of pad sensors, from which features of the performance can be extracted. *Perform* is responsible for predicting when the next musical event will take place and scheduling the accompaniment to happen in synchrony with the human performer. The third process is the ability of the computer to *learn* timing data for the piece from rehearsal.

By analogy with the anatomy of a human performer, Vercoe defines three temporal regions prior to the playing of a note. At a distance in the future, the note is merely a score event. As the moment for its performance approaches, it is drawn into short-term memory, when it *can* be given chronological definition. As this moment of time approaches, the system schedules the necessary command to play it and "*must*" commit to the performance of the note.

The *Learn* process works by estimating the local tempo and calculating the mean and standard deviations of the rhythmic aberrations for every note from the rehearsal performances. Tempo matching then takes place to the *mean-corrected* events and the inverse of the standard deviation is used to weight the importance of these events in determining the tempo. The result of this training mechanism is that the accompanist is far more

sensitive to the soloist's musical expressions resulting from distortions in the tempo and note onset times. The requirement for the *Learn* process is motivated by observations on the nature of musical aesthetics and interaction:

> "Computer performance of music can easily demonstrate that strictly synchronous behaviour lacks much of the information we routinely seek from live performance. It is as if the musical score acts as a carrier signal for other things we prefer to process. Much of this information derives from discrepancies between individual players. The degree of synchronisation will vary ...[and] a performer will seek an aesthetic way of adapting, so as to preserve the integrity of his own line. ...We have here, in effect, a loosely-coupled system of performance and control, whose parameters depend on the topology of the score involved." [VP85]

Vercoe and Puckette perform score following by assimilating Dannenberg's approach to the problem into the Synthetic Performer and incorporating useful information learned from rehearsals. They assign a cost to every mapping between the performance and the score by penalising missing, extra and wrong notes and also metrical deviations from the estimated local tempo. Their dynamic programming algorithm works by remembering four least costly theories which can be used to calculate the best theory for the next note in the score. Quantitative errors, such as early or late notes, are accommodated using an unspecified averaging process. Once sufficient statistical data has been learned from rehearsals, the Synthetic Performer anticipates the performer to provide a robust yet sensitive accompaniment.

Grubb and Dannenberg [GD97] were the first researchers to formalise the problem statistically using probability distributions for the performer's location in the score. Assumptions of independence are required to simplify the calculation over probability density functions. Raphael [Rap99] also approached the problem from this perspective in his Music Plus One

(MPO) system, making use of the hidden Markov model (HMM), successfully used in many sequential analysis tasks such as speech recognition [Rab89] and landmine detection [GM99], to model the note transitions within a piece. He subsequently developed the architecture of the system using Bayesian Belief Networks [Rap01][Rap02]. These methods require training so that the system encodes transition probabilities and expectations learned from rehearsal.

The HMM works by assuming that the observed states, derived from the processed audio of the performance, are created by a hidden sequence of states, resulting from the player's movement through the score. This assumption in the architecture of an HMM has made it a popular method of tackling the problem of score following. In Raphael's model, the states used in the hidden layer represent the attack and steady-state parts of each pitched note of the scale and a rest state is included for when the soloist is not playing. Each note within the score is modelled as a series of states which are then chained together.

Raphael then models the tempo and duration of each note as two random variables. The system trains on rehearsals to learn how the tempo tends to fluctuate within the piece and to what extent each note has an early or late onset relative to this underlying tempo. By training the system on previous renditions of the piece, the Music Plus One system is sensitive to the rhythmic variations of the musician. Raphael has extended the system to use real audio accompaniment of a recorded orchestra by time-stretching the accompaniment to synchronise with the soloist.

Following Vercoe, Raphael delineates *listen*, *perform* and *learn* processes for the system. Where previous score followers had used pitch-to-MIDI conversion, the *listen* component of Raphael's system processes monophonic audio to form a vector of features, including the energy of the signal and the presence of individual notes computed via a Fourier transform. Music Plus One used the phase vocoder to time-stretch audio when generating accompaniment and using orchestral accompaniment, Raphael has successfully demonstrated his system with instruments including oboe and violin. Several audio and video examples are viewable on Raphael's

webpage [1].

IRCAM have developed a set of Max patches, *Suivi* [OF01] [OLS03], which performs score following from audio input and shares many similarities with Raphael's MPO project. They use a low-level HMM to recognise the sequence of features which correspond to the attack, sustain and decay of a note. The system then analyses the score to generate a higher-level HMM which calculates the probability transitions between states which correspond to notes in the score. The HMM is dynamic in the sense that the set of states used changes as one progresses through the score. For a particular point, approximately twenty notes might contribute information to the transition matrix and the number of states might be around 200. *Suivi* tends to be used on monophonic instruments at present since pitch detection is much more reliable when there is only one note to detect. A 4096 sample frame FFT, with a hop size of 512 samples, is used as input to the system which extract features from the Fourier transform of the audio. A note is more likely to be detected once it is present in the central section of the window and so the resulting latency is up to 2000 samples or 45ms. This delay is analogous to the kind of delay one finds naturally in a large room and the system has been successfully used to provide automated electronic parts for several classical pieces which blend natural and electronic sounds such as *En Echo* by Philippe Manoury and *Explosante-fixe* by Pierre Boulez.

Recently Arshia Cont and IRCAM have developed the *Antescofo* system [Con08a] [Con08b] for anticipatory score following, whereby the predicted time until future events is specified and a running tempo and bar location is provided. The tempo model uses Large and Jones's oscillator [LJ99] to update the tempo estimate on the basis of observed IOI's from the score following module. This modelling of tempo in a score follower allows the scheduling of future events so that the system may no longer be merely reacting to observed notes in the score, but also anticipating them. Support for Ante-scofo has been added into Keith Hamel's NoteAbility Pro system for score representation [LH07], thereby providing a

[1] `http://xavier.informatics.indiana.edu/~craphael/music_plus_one/`

additional functionality through visualisation of the score in real-time and creating an 'Integrated Interactive Music Performance Environment'.

2.1.2 Automatic Accompaniment Systems

Whilst score following systems are a class of automatic accompaniment system, there is also an alternative class of automatic accompaniment systems which are not reliant upon scored music. Such a system might choose to match performances of the same piece which adhere to the same musical structure. For instance, a blues or jazz piece can be loosely defined in terms of the underlying chords and this structure can be successfully followed without there existing a score which specifies the musical content to the level specific pitched notes.

Simon Dixon's MATCH Toolbox [Dix05] uses Dynamic Time Warping (DTW) to create a mapping between two performances of the same piece. Audio is pre-processed by taking the half-wave rectified (i.e. positive) spectral difference to emphasise salient points in the music and uses dynamic programming with an appropriate distance function between two vectors. This system offers a strong alternative to the explicit note modelling used by score-following systems.

2.1.3 Generative Systems

Other interactive systems have been designed around the paradigm of improvisation. The Continuator by François Pachet [Pac02] uses Markovian techniques to interact with a pianist in a novel way. Pachet was influenced in this by composer David Cope [Cop96], who used Markov models to discover new musical phrases that were coherent with his own style. It builds a database of patterns played by the musician and indexes all subsequences of the input. When the musician stops playing, the system continues the phrase by using the transitions from the longest subsequence matching the input to continue the phrase. The probabilistic matrix that is created is fully determined by the user's input. However, due to Markovian processes which generate the continuation, the output has the appearance of

being a creative extension within the style of the musician. The Continuator has several modes. In the first, 'Autarcy', the system has no memory and progressively trains on input to learn the musical style of the performer. In the 'Virtual Duo' mode, the 'memory' of another musician is used as the transition matrix for the system, with the result that phrases are completed in the style of the previous player. Pachet describes the 'Aha' effect that has systematically surprised musicians on the realisation that the Continuator has started to play in the same style [2].

Another system designed for improvisation is GenJam by John Biles [Bil07], which uses genetic algorithms to modify a population of individuals, in this case musical phrases or 'licks', which are played over a set sequence of jazz chords. Each individual is classified according to its suitability as a solution to a problem, which in this case is the aesthetic appeal of the phrase as a musical improvisation. This suitability rating is used by the algorithm to determine the individual's evolution, both its own survival in the population and how it is combined with other individuals to create new individuals. Since it is hard to carry out this evaluation automatically, Biles acts as the arbiter of fitness within the evolutionary process. By the fifth generation, the improvisation begins to achieve some aesthetic success with respect to tonality and rhythm. He relates his experience of this process [Bil94]:

> "The first few generations of a training session are quite numbing for the mentor. Fitnesses are almost all negative, melodic intervals tend to be large, and the frequency of 'nice moments' is very low. Sooner or later, though, a few pleasant licks begin to emerge, and one or two solid phrases tend to appear by the fourth or fifth generation."

Biles modifies the generator of the sound to bend into the notes and introduces variations within onset and duration time. This has the effect of "humanising" the part by introducing deviation from the timing of a strict interpretation. A live demonstration of GenJam was given at the

[2] Video of the Continuator in practice is viewable at `http://www.csl.sony.fr/~pachet/Continuator/index.html` (as viewed 7th May 2009)

ICMC'98. Videos of the system are also viewable on the internet [3].

Blackwell and Young created an interactive system, Swarm Granulator [BY04], which creates musical events from the movement of swarm-like particles in a virtual space. This system is inspired biological systems which exhibit self-organisational properties despite the lack of central control. Each member of a swarm obeys a simple set of local rules inferred from nature that govern its behaviour:

1. Move towards a given attractor
2. Try to move at the same speed and in the direction of your neighbours.
3. Do not collide with your neighbours.

The system is designed to act as a mechanism to integrate the computer into 'free' improvised music where traditional musical structure and form are eschewed in favour of the dynamics of social processes. Attractors are created within this space by analysis of the audio from musicians and other swarms and the particles repel those close by them, so that globally, as a swarm, they gravitate towards the attractor but do not coalesce into one another. The particles' behaviour around the attractors has a musical influence over the actions of the performer, hence bringing about an interaction between the human improviser and the computer generated events. Audio of the improvised performances featuring the Swarm Granulator are available via the internet [4].

2.2 Music Psychology

Beat Tracking has been the subject of investigation for many years. Many commentators, e.g. [Ros92] [Dix05] [Hai06] [KEA06], have observed that it is simple for a person to tap their foot in time to music at locations where there is a perceptual beat. It has, however, been surprisingly difficult to enable a computer to do the same task. Partly, the difficulty is due

[3] `http://www.it.rit.edu/~jab/GenJam.html` (as viewed 7th May 2009)

[4] `http://www.doc.gold.ac.uk/~mas01tb/SwarmGranulator/swarmgranulator.html` (as viewed 7th May 2009)

to the fact that the beat is a perceptual construct, the result of complex, parallel processing in the brain [TLO02], which relies on the mutual influence of several levels of cognitive processing. Constructs such as pitch and beat are generally recognised to be mental phenomenon, products of our interaction with the world, so that although they are caused by external stimuli, these terms do not necessarily correspond with something that can be simply measured anything from the signal [Sch98] [Han89]. In addition, the mental processes may make use of top-down processing, whereby information is fed back to inform the processing at lower levels.

Meyer [Mey56] identified expectation as key to the generation of meaning through music and proposed that deviation from our expectations induced emotion in the listener. Meyer's theory is derived from the law of affect, a proposition from psychology, stating that "emotion is aroused when a tendency to respond is inhibited." By "tendency", Meyer signifies a pattern reaction, or set of mental or motor responses, that unless inhibited, follow a previously ordered course. Whereas in everyday life, the resulting tensions often remain unresolved and subsumed by the succession of irrelevant events, in art, the relationship between tendency and resolution is made explicit and the tendency thereby derives meaning.

> "In music, ... the same stimulus, the music, activates tendencies, inhibits them, and provides meaningful and relevant resolutions."

The "tendency to respond" may be conscious or unconscious, but the more automatic the response, the less it is brought to the conscious mind. Expectation in music gives rise to states of suspense where the resolution is in doubt and there is a subsequent progression from tension to release. Our expectations are both structural and temporal. With respect to tempo, the beat is a product of our perception of regularity underlying the music, defining the moments at which we *expect* beats, salient musical notes and chord changes to happen. Dannenberg [Dan05] describes the paradox involved in the problem of beat tracking: "If everything depends on everything else, where does one start? If perception is guided by expectations, will we fail to perceive the truth when it is unexpected?"

Generally there are thought to be two main components to the problem of beat tracking. The first is beat or tempo induction, which estimates the tempo. Dahl [Dah05] enumerates three types of tempo in music performance : the mean tempo, averaged across the whole piece of music, the main or prevailing tempo and local tempo that is maintained only for a short time. In some musical genres, tempo is constant enough for these three categories to merge. The application of tempo induction across a piece relies on the mean tempo being a good approximation for the main and local tempos. Beat tracking is the subsequent problem of placing the beats close to the perceptual beat as agreed by humans. The problem of bar boundary location is an extension of the problem of discerning the phase by classifying each beat with respect to its position in the bar and thereby the time signature and meter.

2.2.1 Temporal Organisation of Sound

Cooper and Meyer [CM63] proposed definitions of three different modes of temporal organisation: pulse, meter and rhythm.

Pulse

"A pulse is one of a series of regularly recurring, precisely equivalent stimuli. Though generally established and supported by objective stimuli (sounds), the sense of pulse may exist subjectively."

Meter

"Meter is the measurement of the number of pulses between more or less regularly recurring accents." Meter thereby specifies regularity of perceived structure within a piece of music. Cooper and Meyer note that "When pulses are counted within a metric context, they are referred to as beats. Beats which are accented are called strong; those which are not are called weak." Thus all meter requires the existence of an underlying pulse.

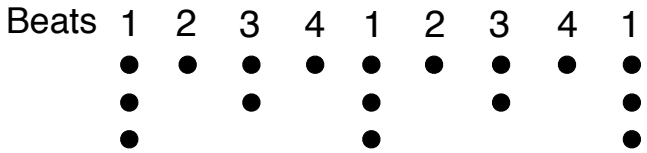

Figure 2.1: The Generative Theory of Tonal Music's metrical structure of a bar, featuring alternating strong and weak beats.

Rhythm

"Rhythm may be defined as the way in which one or more unaccented beats are grouped in relation to an accented one."

Lerdahl and Jackendorff's Generative Theory of Tonal Music [LJ83] (GTTM) has had considerable influence on researchers in beat tracking. Their approach is derived from musicological studies of classical music and extends to many genres of western music. The theory is not concerned with the intricate timing of a performance, but operates at the abstracted level of a score or quantised representation, thereby discounting the interpretative process required to correctly infer metrical position from expressively timed events. The GTTM, by analogy with linguistics, seeks to define a grammar which describes the musical intuitions of an experienced listener. A generative grammar attempts to describe an infinite set (of possible musical pieces) by formal finite means. Whereas linguistic grammars employ *well-formedness rules* that define whether a given string is a possible sentence, music is not referential or tied to semantic meaning in the same way, and thus *preference rules*, which do not feature in linguistic grammars, play a more important role by indicating which of the structural definitions of a piece correspond to the intuitions of the musical listener.

The GTTM defines grouping structures, which express the segmentation of a piece into motives, phrases and sections. Meter is defined as the hierarchical structure emerging from the occurrence of alternating strong and weak beats, with different periodicities at each level. Lerdahl and Jackendorff provide rules as to how these structural levels operate, so that a beat at one level must be a beat at any lower or weaker level.

This generates a metrical hierarchy as shown in Figure 2.1. This accords with our conventional understanding of meter and with the definition proposed by Cooper and Meyer. Jones and Boltz [JB89] proposed a theory of dynamic attending which concerns regularities in the periodicity of our attending to events. Temporal hierarchies refer to time structures in which the distributions of temporal markers are consistently related by either ratio or additive transformations. Meter involves a simple integer ratio relationship between two time levels which generates the same periodic, hierarchical structure as found in the GTTM.

Although we do not require the full power of the GTTM for an explanation of meter and rhythm, it might play an important role in future interactive systems if it can be used to interpret the musical structure of a piece in terms of phrases and groupings. Time span reduction seeks to explain the positioning of pitched notes by reference to structure and underlying key. Thus, given an interpretation of musical structure, an interactive system might use the GTTM to create musical variations that will satisfy the constraints expected by listeners.

2.2.2 Synchronisation as a Psychological Phenomenon

We will be looking at the approaches taken by previous beat tracking systems, but we can also learn relevant facts from studying the behaviour of a highly efficient and accurate beat tracker: *the human*. Tapping tasks have played an important role in the investigation of sensorimotor synchronisation (SMS) [Rep05]. Since we wish our system to rival a human's sensibility to microtiming deviations within the beat, tapping tasks provide experimental data on how this simple feat is accomplished that may not have been anticipated.

When subjects are asked to tap in time with a metronomic pulse, a subliminal local change (between 0.8% and 2% of the period) made to a single interval in an otherwise isochronous sequence results in rapid phase-correction despite the error being below the perceptual threshold [Rep00]. A further experiment [Rep01] contrasts synchronised tapping with free or continued tapping, where the external stimulus is removed, and it is

assumed that the subject will continue to tap at the period of an internal timekeeper or oscillator. When a pulse change is made to a single interval, free tapping continues at the rate of the original period. This experimental evidence supports the two-level timing model, first suggested by Wing and Kristofferson [WK73], also found in Mates [Mat94] and Vorberg and Wing[DW96], which posits separate mechanisms for period and phase, leading to the implication that changes in phase can be made independently of changes in period. A simple form of the model makes a linear combination between an internal timekeeper, controlling the period, with motor-delays which account for phase, so that the interval times are characterised by the equation:

$$I_n = T_n + M_{n+1} - M_n \tag{2.1}$$

where I_n is the n^{th} interval between pulses, T_n is the period of the timekeeper and M_n is the motor-delay. Repp's empirical data from experiments agree with the two-process error correction model, with the added assumption that the period is corrected as the result of conscious awareness of tempo change. The experiments indicate that phase changes are fast, subconscious and automatic, whilst changes in period are slow and may require conscious recognition of the change. Experiments suggest it is possible to achieve synchronisation via changes in phase even in a situation that requires a step-like change of timekeeper period [SVS00]. In this case, what appears like rapid adaptation of the tapping period to a small, undetected tempo change is in fact rapid internal phase correction. If humans tapping to the beat are used to making rapid, subconscious adjustments to maintain phase, then given that our paradigm is a human tapping to a beat, we may need to incorporate a comparable mechanism into our system. Repp suggests the phase-correction process has access to more accurate timing information than our conscious decision processes [Rep00].

Phase relationships other than in-phase or anti-phase are difficult to maintain [Rep05] and participants will synchronise with distractor sequences even against their will. The lower limit for SMS is determined by the speed at which the participant can tap and corresponds to an inter-tap

interval of 150 to 200 ms. However, if the tapping ratio is less frequent (every two, three or four beats), then this level is lowered to 100 to 120 ms [Rep03]. Below this limit, participants are unable to maintain synchrony and drift in phase.

An estimate of limits of human error can be gained through measurements of the asynchronies between participants' taps and the isochronous pulse. Pressing and Jolley-Rogers [PJR97] contrasted measurements from a trained percussionist and pianist with a non-musician, recording standard deviations of approximately 2% of the inter-tap interval for the trained musician and 4% for the novice. Pressing and Jolley-Rogers also measured statistical data when the trained percussionist was required to keep a steady beat with no reference pulse. They observed a slow rise in tempo at 750 ms inter-tap interval (ITI), corresponding to a tempo of 80 BPM, whilst this drift was diminished but still present at a faster tempo of 240 BPM. Hence, with relevance to the kind of musical behaviour that we wish to interpret, we can expect that drummers will naturally exhibit drift when their timekeeping is determined by an internal clock.

2.2.3 Task Description

Toiviainen and Snyder [TS03] describe how a complete model of pulse finding that can account for SMS would need to account for multiple behaviours, including "the extraction of one or more periodicities from the musical signal, the determination of which periodicity is most appropriate for tapping, the generation of motor outputs that correspond to the period and phase of the most appropriate periodicity, and the continual adaptation of period and phase of motor output to compensate for timing variability in the stimulus and in the synchronising mechanism".

Whilst this description focuses on explaining the task in relation to human psychology and physiology, beat tracking systems attempt to simulate the pulse finding task algorithmically and the decomposition of requirements still applies. Thus an automatic beat tracking process can be deconstructed as consisting of (mean) tempo estimation, initialisation of

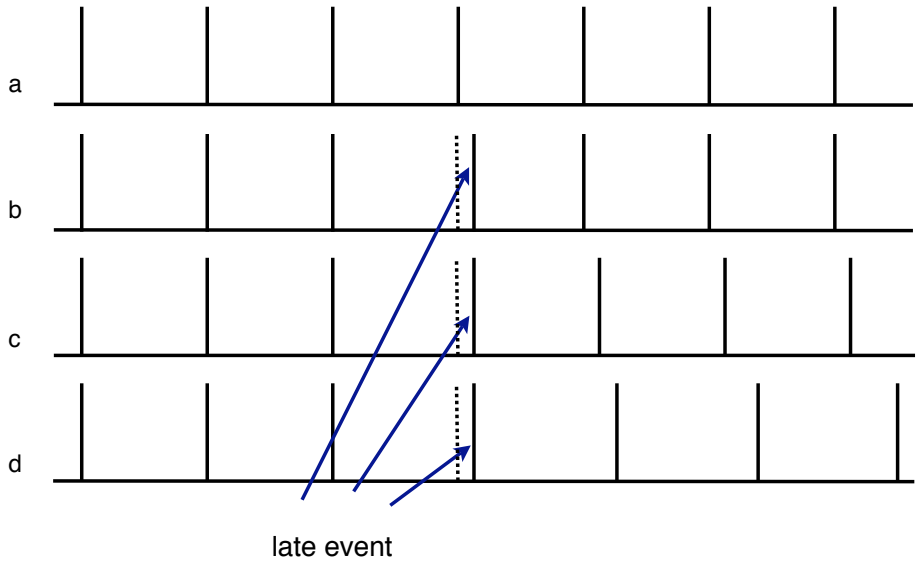

Figure 2.2: Four examples indicating (a) constant tempo, (b) an expressively timed event (c) a local tempo change and (d) a global tempo change. After Gouyon and Dixon [GD05]

period and phase, and adaptation of period and phase throughout the signal. Whereas non-causal algorithms have access to the full signal before placing estimated beats, a real-time system without prior representation of the signal structure will have to operate causally, making use of information solely from the signal's past at any stage.

2.2.4 Tempo, Timing and Causality

We will now introduce some terminology in order to illustrate different variations within timing structure. Figure 2.2 shows a series of examples from Gouyon and Dixon [GD05], displaying a variety of timing deviations within a sequence of regularly spaced events which we can consider as onset times. In the top line, (a), the pulse is isochronous and regular. The second line, (b), is an example of *expressive timing*, where the middle event happens late relative to its expected time, but future events are not affected. The event is thus displaced in time locally with no effect on the tempo. The third line, (c), shows a *local tempo shift*, where the event is displaced but the tempo remains constant, so that future events are

also displaced relative to (a), but the underlying tempo in (c) remains the same. The fourth line, (d), shows a *global tempo change*, where the inter-onset interval has increased by a small factor. Gouyon and Dixon make the observation that a major difficulty confronting beat trackers is that the dimensions of tempo and timing have been projected onto a single dimension of time. Thus, any sequence of events could be represented as a series of tempo changes, but we are seeking the most parsimonious representation in which tempo and phase changes are minimised.

This diagram also illustrates an important point regarding any causal system. At the point of displacement, (b), (c) and (d) have all received identical information. A non-causal or offline tracker can use future information to decide which type of event has occurred; but for a causal or human tracker, without prior information such as that learned from rehearsal or exposure to cultural trends, the only way to interpret the event at the moment of its occurrence is using past observations, and yet, for these three cases, this information is the same. If a real-time system delays its reaction, in order to interpret the event in light of future information, then it is prone to being unresponsive. It is also the case that 'no reaction' has implications from the point of view of being a decision made by the system and reflects a belief that the prior tempo hypothesis is still optimal. Otherwise, we must choose between the most likely interpretation, given our prior assumptions and knowledge from observation, or we may choose a compromise path between interpretations.

2.2.5 Real-time and Predictive Beat Tracking

Dixon [Dix01] makes the distinction between *predictive* beat tracking and *descriptive* beat tracking. His BeatRoot system aims to perform descriptive beat tracking, indicating where the beats *actually* fell, and this is motivated by its application as a tool for musicological analysis. Transitions can be sudden between successive beats when there are changes in timing and tempo which surprise the listener. On the other hand, a predictive beat tracker models perception, predicting the listener's *expectation* of beat times using causal algorithms and thus smoothing transitions.

Causal beat trackers have to make predictions about expected beat locations on the basis of what they know at the current time. For real-time automatic accompaniment, the beat tracker inevitably decides the tempo *ahead* of time and in the systems we are considering, it can only change future predicted beat locations via a change in tempo. In order for this to be musically acceptable, we may wish to impose a limit on the quantity of change that can be accommodated.

The distinction between predictive and descriptive beat tracking can be emphasised by a simple example. Supposing a predictive real-time tracker experiences a global tempo change, where the tempo slows by a given factor, then the predicted location of the beat has already been passed by the time this point is reached. So the required change of tempo to synchronise the next predicted location with the following beat is even greater than the global change that has been made, since it must also accommodate the current error. This assumes that the system can register such a tempo change immediately and reliably. In practice, averaging of some form is required for stability, and yet the predictive system would have to make up all the incremental errors between expectation and observation that have been encountered before recognition of the tempo change.

Reactive systems can be designed that respond to incoming events after a small amount of detection latency. Early score following systems, such as those be Vercoe [Ver84] and Dannenberg [Dan84], recognised events in the score and then cued their response. Provided this event detection was within an acceptable time limit, this method could adequately serve to generate automatic accompaniment and create an illusion of synchronicity. Vercoe [VP85] enhanced the performance by incorporating a learning mechanism that predicted where future events would fall and scheduled events a short moment ahead of time. Raphael has also developed a system that learns the characteristic tempo changes of a piece and this has been used to provide automatic accompaniment to scored monophonic music. The key development here is the switch from a *reactive* system, which triggers an event from detection another event that ought to be simultaneous, to *predictive* systems, that can predict the future adequately enough so that future events can be scheduled from observations in the recent past.

This is, after all, how humans behave when they create music.

Vercoe [BD86] found that "acceptable response appears to require a close model of the physiological processes involved in actual human performance, including score-lookahead, gradual focus on a forthcoming event, then an anticipatory action decision." Collins [Col06] points out that humans have a larger latency than computers when required to identify notes, and yet we have learned to perform highly complex musical structures in group ensembles. Lower latencies alone will always be insufficient to emulate true musical interaction so long as purely *reactive* systems are used and, particularly when events have slow attacks, they need to be triggered *before* the perceptual onset time. He makes the case that musical systems need to emulate the predictive power of humans if they are to succeed in truly interacting and improvising. The challenge is not to reduce latency in the detection process to near zero, but to translate these detections into reliable predictions when scheduling future action.

We will describe previous approaches to the problem of beat tracking, both causal and non-causal, but first, we shall investigate some of the qualities specific to the signals we expect to encounter in creating a beat tracking system for drums.

2.3 Characteristics of Drum Signals

In a rock or pop band, the rhythm section is the union of bass and drums which provides the metrical structure on which the others play. In order to provide automatic accompaniment for a band, we need to synchronise our system to the meter of the live rhythm section. Although 'the beat' is generated by several instruments in unison, our choice has been to lock to the drums, on the basis that strong drum events like kick and snare are the clearest indicator of where the beat falls. By tracking drums in real-time, we can analyse any other instrumental part with respect to the metrical structure of the drum pattern. Before describing the system we have developed, we shall look at some specific qualities of the drum signal and the styles of playing that characterise the instrument.

At the core of most drum beats is the interlocking pattern created by

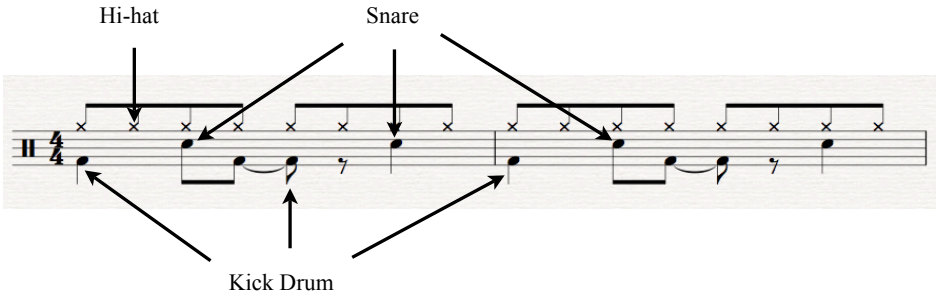

Figure 2.3: Basic rock beat with conventional drum notation. The hi-hat pattern is a sequence of regular "eighth notes", which recur at the tatum level. The snare is present on the backbeat, 'two' and 'four', with kick drums on the 'one' and 'three'.

the kick drum, the snare and a cymbal pattern, played on the hi-hat or ride cymbal. For a right-handed drummer, the snare and hi-hats are close together on the left-hand side. Due to the fast transients and high sound pressure levels in a snare drum signal, a dynamic microphone, commonly a Shure SM57 (e.g. Hirsch and Heithecker [HH06]), is recommended for both the studio and live environments. A dedicated kick drum microphone is placed either inside or in front of the drum head and a dedicated microphone for the hi-hats may be used if the venue is sufficiently large. This microphone set-up allows reasonable separation between the kick drum and the snare, although there may be some 'bleed' between them, whereby the kick drum is picked up by the snare microphone and vice-versa. Much energy from the hi-hat will often be present on the snare drum microphone, but when the snare is hit hard, it is the clearest acoustic signal present. In rock and pop drums, the playing style creates distinct events, such as snare hits, whereas in jazz, brushes may be used which create a more continuous sound on the snare.

Performers use variation in tempo as an important means of expression. In classical music, a *rubato* section may slow dramatically, creating a sense of anticipation for the next events. Their sense of rhythm is disrupted, so that the listener searches for a new tempo, a new organisational structure within which to make sense of the sound. When drums are played within a band, since an ensemble of musicians is involved in creating a collective pulse, there tends to be less dramatic variation in tempo. However,

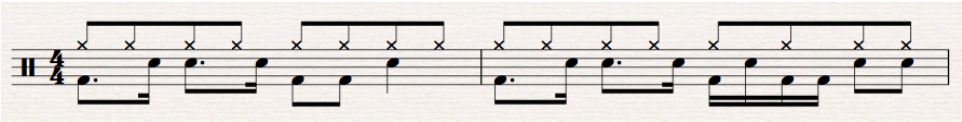

Figure 2.4: Syncopation example, after Tommy Igoe.

rhythmic deviations can still occur, but to a lesser degree. Onsets may be placed either early or late, either locally as stylistic deviations (expressive timing) or constituting local tempo change (described in section 2.2.4), and there are variations of the global tempo. In addition, drummers are prone to introducing *fills*, sections of rapid events, often falling on subdivisions of the metrical level, which must be successfully interpreted by the tracker.

Surprise occurs in the form of syncopation (the stressing of normally unstressed beats), changes in pattern, and expressively timed events. In his article for Modern Drummer magazine [5], Tommy Igoe [Igo06] describes this phenomenon:

> "One of the things drummers love about many funk grooves is the syncopation (the shifting of accents) within the pattern. The example [in Figure 2.4] uses a common technique called displaced back-beat. The backbeat in contemporary music is on beats 2 and 4, but here we've displaced the backbeat on beat 2 by moving it one 16th note earlier to the "ah" of beat 1. This displaced back-beat does two very interesting things: It forces the groove out of balance, and it opens up the second half of the bar for numerous rhythmic variations."

Syncopation is thus viewed as a permutation of rhythm by a sub-division of the regular pulse. Temperley [Tem99b] has put forward an extension of the General Theory of Tonal Music to account for the syncopation of melodies in rock music, which appear to contradict the GTTM rules that stresses tend to fall on strong beats. Syncopation can then be viewed as a displacement of a deeper rhythmic structure, where the syncopated event

[5] Available via the internet at `http://www.moderndrummer.com/drum-education.php` as viewed 7th May 2009

happens *earlier* than the strong event to which it corresponds. The theory gains weight from empirical evidence and the observation that syncopation in the other direction, where events are displaced later, is not commonly observed.

Jeff Pressing [Pre02] has tried to characterise the qualities associated with what he terms 'Black Atlantic Rhythm'. These rhythms, shared culturally between America and Africa, have given rise to many forms of popular music: jazz, blues, rock, reggae, hip-hop. These rhythmic styles are principally those that we are aiming at interpreting in real-time. The rhythmic devices used by such rhythms all rely upon "the support of a firmly structured temporal matrix", defined as a "groove". This is characterised by the perception of a regular pulse with a subdivision structure, the perception of a longer time cycle, and effectiveness in entraining the human body to synchronise response and movement. Waadeland [Waa01] associates the quality of "groove" with a rhythmic phenomenon, resulting from the conflict between a fixed pulse and various timing accents played against it, or resulting from the "musician moving in non-metronomical ways".

Pressing enumerates several rhythmic devices which "build on the groove". These include syncopation, displacement, off-beat phrasing, polyrhythm, hocketing (an interlocking pattern shared between multiple instruments) and swing. The groove is the template through which surprising variations in form can be understood. In addition to syncopation, where the displacement of an event is by sub-divisions of the main pulse, there can be displacements at the microtiming level. In jazz, funk and latin music, the quality called "swing" relates to the perception of "groove" by an uneven division within a beat [Iye98], whereby the two eighth-notes constituting a quarter-note beat no longer need to have durations in a 1:1 ratio. Whilst Jazz swing can be conventionally notated by dividing the beat in the ratio 2:1, corresponding to the first eighth-note being extended to the duration of two triplet eighth notes, in practice this ratio varies considerably. Friberg [Fri99] demonstrated that this ratio decreases at faster tempos. Berliner [Ber94] relates that for jazz music "within the realm of beat subdivision, myriad nuances of phrasing in between an even

eighth-note subdivision feel, a dotted-eighth and sixteenth-note feel, and a triplet eighth-note feel are associated with the dynamism of swing." Freeman [FL02] reports that the analysis of two jazz recordings of the same song, "It Don't Mean A Thing If It Ain't Got Swing", shows idiosyncratic differences between individuals. Drummer Gene Krupa consistently playing the swing at 62% and Buddy Rich playing it at 69%.

The main argument for the musical, rather than random, nature of microtiming deviations rests on the consistency with which deviation patterns are repeated in the same or similar musical contexts [McG06]. Bilmes [Bil92] conceives of an *event shift* function, required to express the rhythmic deviations observed in African and African-American music, which measures the expected timing deviation of events at a particular bar position and tempo. By analysing James Brown's "Funky Drummer" by hand using accurate audio editing software, Freeman [FL02] found that the drummer, Clyde Stubberfield, consistently played the snare on beat two late, with a mean lag of 2.8% of the beat period (17msec). Analysis of Cuban drum music by Alen [Alé95] also identifies consistent delays on specific beats that fall within a 30msec window.

This notion of displacement is widely acknowledged by drummers, who talk of playing "behind the beat", "in front of the beat" and the creation of "a pocket", which is characterised by a very steady tempo and a lack of ornamentation. Steve Anisman [Ani97] describes the notion of placement in an article for The Modern Drummer:

> "Every member of the band gets to make a decision as to when they will play their part, in relation to that precise moment [the beat]. Some people like to play their parts behind the beat. This does not mean that the player is playing slower than the rest of the band. The player is playing in perfect time, and his pulse matches the pulse of the rest of the band precisely. It is just that this players 'pulse clock' got started a millisecond or two after the first note of 'the beat', and every note that this player plays is a little bit late, technically."

Anisman describes the phenomenon of expressive timing or microtiming as it occurs in rock music. Although the shifts in timing are very subtle, only several milliseconds, it has implications for all beat tracking systems seeking a level of accuracy attained by humans. Most assume that observed peaks in the detection function must correspond to "the beat" when they are approximately aligned. In offline trackers, accuracy is required to a lesser extent, so this may be a reasonable assumption to make. However, the beat is a perceptual phenomenon arising from the mutual interaction of musicians and as such there may not be any definable quality in the signal that corresponds precisely to it. There are several difficulties to conducting a study into microtiming. The "beat" needs to be annotated, but there are limits as to the degree of accuracy that can be achieved. In addition, the study requires the investigation of the relationship between event onsets of different instruments. Onset detection functions can be used to transform note onsets, events with duration, into onset times, which do not. Since such a study would be concerned with relationships in microtime, then care would be required to ensure that bias is not introduced through the analysis process.

When a drummer plays "behind the beat", they are placing events a few milliseconds behind the perceptual beat which has the effect of making the other members of the band appear to "drive" the song forward, since other instruments will be sitting ahead of the drums. This characteristic is often attributed to John Bonham of Led Zeppelin. Examining the first bar of the classic "When The Levee Breaks" (see Figure 2.5), we can observe that some beats are placed later than the expected location that would be implied by strict equal division of the bar. However, the beats do not always fall late in the same way. After the first intro bar, the snare on the 'two' is relatively accurate, but it is the syncopated hits, which are placed the eighth-note before the 'three' (bar marker 1.4.3 in Figure 2.5), that show the most deviation, happening late relative to the metrical grid. This is an example of Temperley's [Tem99b] conception syncopation in rock, where it is as if the pattern has been stretched to bring the 'three' an eighth-note early. By placing it marginally late, this may lead to an increase in suspense between the anticipated 'three' and its early

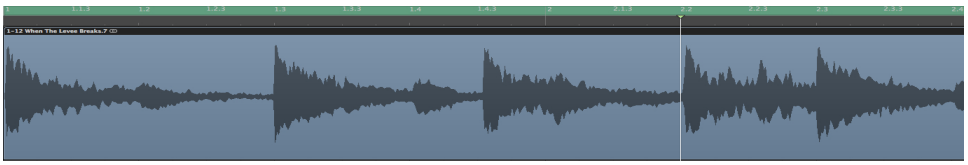

Figure 2.5: The first bar of 'When The Levee Breaks' by Led Zeppelin. The syncopated beat at location 1.4.3 and the snare hit at 2.2 both happen slightly "behind the beat".

syncopated placement. There is also a change in tempo over the first few bars, from 141.2 BPM on the first intro bar, to 142.0 BPM on the second and 142.8 BPM on the third bar, which then remains relatively consistent over the next few bars. Our aim for an accompaniment system is to find a balance between responding to such subtle shifts in tempo, whilst not over-reacting to the kind of timing displacement that we observe in classic rock drummers such as Bonham.

Gouyon et al. [GFB03] observe that in rock and funk, drummers often play quarter-notes slightly "behind the beat". This is visible in the energy from Bonham's hi-hats, visible on the second eighth-note of each beat, where this energy is not obscured by kick and snare events, which are consistently late. These events, a repetitive pattern of eighth notes, support the notion that the division of the beat into eighth-notes is often uneven with the latter eighth-note "swung" slightly late, giving the resulting pattern a certain "feel". Vijay Iyer, jazz artist and professional composer, makes observations relating to regular microtiming deviation of the backbeat in African-American drumming in his PhD thesis:

> "The curious point about the backbeat in practice is that when performed, it displays a microscopic lobsidedness. If we consider the downbeat to be exactly when the bass drum is struck, then the snare is very often played ever so slightly *later* than the midpoint between two consecutive pulse. Often musicians are aware of it to some degree, and they have a term for it: the drummer is said to play 'in the pocket'. ... A skilled musician or listener in this genre hears this kind of expressive microdelay as 'relaxed' or 'laid back' as opposed to 'stiff' or 'on top'."[Iye98]

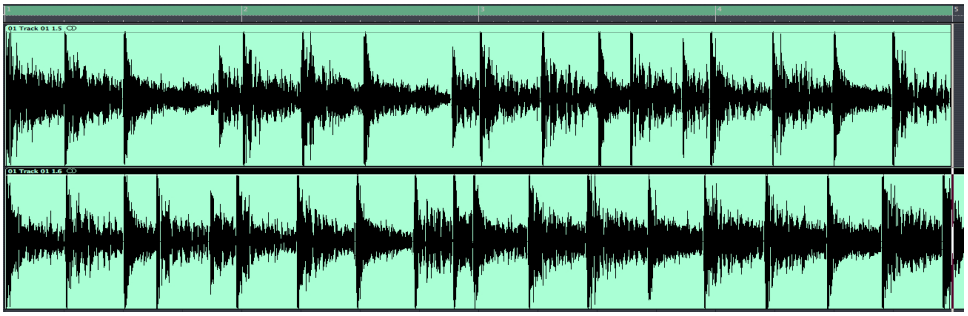

Figure 2.6: The first two sections of a drum take by Led Zeppelin's John Bonham. The second two-bar loop (bottom) is actually marginally faster than the first (top) as can be seen from the waveforms.

In Figure 2.6, whilst the first snare drum is precise, the snare hit on the 'four' (beneath marker 2.3 in the diagram), is visibly late by as much as 30ms. However, there is significant movement within the underlying tempo, even during this short segment, which makes it difficult to distinguish expressive timing from tempo variation and motor error.

2.3.1 Studio Practice and Sequencing: Playing to the 'Click'

One common attribute of sequencing software programs such as Logic, Pro Tools, Cubase and Ableton Live, is that they provide the option for a band to record to a 'click track': a highly accurate regular pulse determined by the computer's CPU. Many commercial songs are recorded to the click, or the "grid", to intensify the steadiness of beat. This has been a common practice since the introduction of drum machines in the 1980's. The use of the click has been extensive over recent years, with many indie guitar bands also making use of it to tighten up their sound. Nirvana's "Smells Like Teen Spirit" was not only recorded to click, but the preferred mix by engineer Andy Wallace featured extensive "tweaks" of the drums parts which are aligned to their ideal metronomic location, removing any expressive timing from the performance [Izh08]. The popularity of the click does support the notion that for some types of music, it is preferable for there to be no global or local tempo changes.

The result is that many songs on radio sound exciting for the fact that they are 'tight' and feature many computerised sounds, such as samples

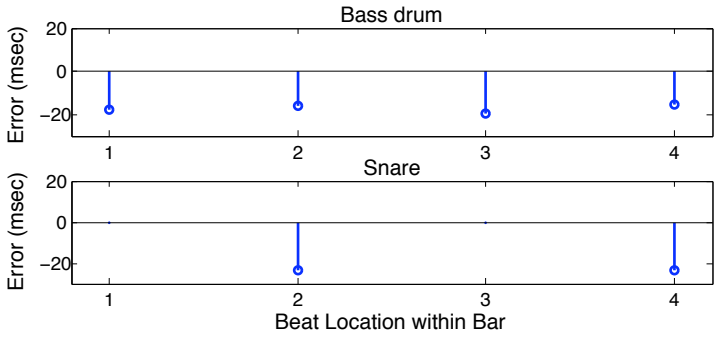

Figure 2.7: Placement of the bass drum and snare relative to the click track by David Nock on a song, 'Ride', recorded to click track.

and synthesizers, positioned perfectly with respect to the musicians; but for some listeners, the songs lack a quality that older recordings have, an different form of excitement, where the musicians are interacting without the need to accommodate a metronome into the performance. The click denies any tempo fluctuation and so when it is used, the listener can quickly predict where events will be located temporally and there is no denial of the anticipation, no surprise.

Drummers often practice to a metronome in order to gain insight into their sense of timing relative to an absolute. When practising to a click, Matt Ingram, session drummer, described how he would aim to "bury the click", to play in synchronicity to the point where it became inaudible to him due to the hi-hat pattern and other drum events which masked it. He could only hear the click when his timing had strayed. This statement has two interesting implications. The first is that drummers seek to maintain a constant global tempo, or at least be capable of doing so. The second is that drummers do intend to place some events, such as in this case hi-hats, 'on the beat' with respect to the click. They then learn how the expressive timing of certain events relative to others affects the "feel" of the pattern. It is our aim to interpret the drummer's playing style with respect to a virtual metronome, whilst adjusting the timing of this metronome so that it remains in sync with the perceptual beat.

In order to learn more about the interaction between drummer and the underlying pulse, we examined how a drummer actually plays *to* a click

track in the studio. On the song 'Ride', we analysed the snare and bass drum signals through an onset detector, relative to the absolute position on the grid of the audio sequencer. The relative displacements are shown in Figure 2.7. David Nock appeared to place the majority of bass drum events and almost all snare events slightly ahead of the click. Perhaps in this song, he wanted to convey energy. However, this could also be the result of *negative mean asynchrony*, the effect first observed by Dunlap in 1910 [Dun10], whereby humans consistently tap 20 to 50 ms ahead of a regular pulse. Further investigation would need to take place to conclude whether this is a regular effect observed in drummers.

Whilst this is not intended as a conclusive demonstration that drummers make deliberate expressive timing deviations relative to the click, it is clear that they may not always play exactly on it. It is important that we remain aware of the many possibilities for timing deviation within drumming: syncopation, expressive timing relative to the perceptual beat, expressive timing relative to other parts of the drum kit, local tempo changes such as a "push" on a particular beat and small global changes in tempo, such as when a chorus speeds up slightly. These timing deviations all occur within music that is at a constant tempo and meter. Songs may also feature discontinuities in tempo which must be tackled in the same way that human musicians would tackle them: by mutual agreement through rehearsal.

2.4 Beat Tracking

We shall describe here previous approaches to the problem of beat tracking, both causal and non-causal, and discuss them in the context of creating a real-time system.

2.4.1 Pre-processing

The audio signal first requires pre-processing to create a 'driving function' for the beat tracker. This is either a sequence of discrete onset events

or an onset detection function representing changes within the audio, using features such as spectral difference, energy, high-frequency content or phase deviation.

2.4.2 Oscillator Models

The use of oscillators for beat tracking is motivated by the theory of dynamic attending [Jon76] [JB89], which holds that the perception of rhythm causes an entrainment or synchronisation of internal rhythmic process to an external musical stimulus. The concept of 'entrainment' has existed since Dutch physicist Christiaan Huygens [Huy86] identified how two pendulum clocks moved with the same period and it has subsequently been applied in mathematics and in the physical, biological, and social sciences. In the 1920's, Appleton and van der Pol [AvdP22] showed that the frequency of an oscillator could be entrained or synchronised by a weak signal of similar but slightly different frequency. In 1976, Jones [Jon76] proposed that the organisation of perception and memory is inherently rhythmic in nature, with many perceptual rhythms of different scales of frequency being involved in mental processes that correspond to external stimuli.

> "At each level of a pattern's structure there is a perceptual rhythm that can match its time properties. Thus, a set of rhythms, graded in periods, responds to world structure." [Jon76]

It is known that two rhythms of close frequency can entrain each other so that they come to occur in phase and with the same period. In the theory of dynamic attending, such entrainment then gives rises to expectations in the listener, so that identification of novel tone patterns or familiar speech is more successful when the timing extrapolated from an initial pattern coincides with the observation of new information [Han89]. When there is a regular periodic function, corresponding to increased attention, information occurring at those temporal locations is more easily assimilated.

The theory implies that our recognition of structure within music will be aided by temporal regularity of features which define the auditory pattern. McAuley [McA95] and Large and Kolen [LK94] have investigated the use of adaptive oscillators to model internally-generated expectancies arising from a regular attentional pulse and entrainment to external musical rhythms. Both authors suggest linking several oscillators together in a network, each corresponding to a different metrical level. For a single oscillator, the frequency and phase is adapted by differential equation of the form shown in equation 2.2, and the oscillator will synchronise with an external pulse of similar frequency.

$$\Delta t_x = \eta s(t) \frac{p}{2\pi} \mathrm{sech}^2 \gamma (\cos 2\pi\phi(t) - 1) \sin 2\pi\phi(t), \tag{2.2}$$

where t is the event time, t_x is the expected event time, $s(t)$ is the signal impulse which is 1 only when an event occurs, η is a coupling strength parameter, and γ is the output gain which inversely affects the width of the oscillator. The γ parameter decays toward zero each cycle, so when there are no events the temporal receptive field of the oscillator widens.

Toiviainen implemented Large and Kolen's algorithm in his Interactive MIDI Accompanist [Toi98], with the added observation that perceptually salient notes, with longer durations, should cause a stronger adaptation than shorter notes. The problems for the oscillator model, as highlighted by Toiviainen, result from the absence of metrical representation in the adaptation process. The system is highly dependent upon initial conditions as to whether it synchronises in phase correctly, and it functions better when there is polyphonic input, such as accompaniment, since this provides a more regular rhythmic structure to the input. There also appears to be a trade-off between rhythmic complexity and the ability of the system to tolerate tempo change.

The InTime system [6], first released in 2002, is a commercial implementation of work by Large, which outputs a tempo from MIDI input. If the tempo is controlling an accompaniment, the effect is that the accompaniment will synchronise to the player through phase-locking. One

[6] `http://www.circularlogic.com` as viewed 14th April 2009

stipulation is that the player must listen *to* the accompaniment, which guarantees that the phase is approximately correct. Despite the absence of musical interpretation of the input, in terms of metrical structure, beyond that imposed by the oscillator, the effect is impressive and demonstrates the potential of oscillators for mirroring our own psychology.

2.4.3 Comb Filter Resonators

Scheirer [Sch98] proposes that the rhythmic properties of a musical signal are preserved when calculating amplitude envelopes over six sub-bands and then convolving with white noise. The derivative of the resulting signals for each sub-band is fed into a network of comb filter resonators tuned to appropriate frequencies. Good results were given using 150 resonators, logarithmically spaced over the range 60 to 240 BPM. The outputs of the resonators are examined for phase-locked behaviour and this information is tabulated. The tempo is calculated by summing across the frequencies for the six sub-bands and calculating the maximum value. Whilst a 2Hz click track results in a distinct spike at the corresponding tempo of 120 BPM, when tested with an unspecified example of 'polyphonic music', Scheirer reports that the maximum energy value was only approximately 120% that of the minimum value. Thus, although the tempo can be determined through such a filter bank, there is not always a large distinction between correct and incorrect tempi. Phase is then calculated by examining the resonators corresponding to the frequency of the estimated tempo and examining its "predicted output".

A comb filter with delay T and gain α has a magnitude response:

$$H(e^{j\omega}) = \left| \frac{1-\alpha}{1-\alpha e^{-j\omega T}} \right|. \tag{2.3}$$

The parameter alpha affects how the comb filter behaves with respect to new information. A high value, close to unity, means the algorithm will "lock on" to a beat and not be perturbed by energy in the signal at new periodicities, whereas a low value means that the algorithm is quick to change its estimate. This feature of a trade-off between inertia and reactivity will be encountered in other work on beat tracking. Scheirer

recommends a genre-dependent approach, using higher settings for music with a regular pulse such as rock and pop, and lower values for classical music where the tempo can change rapidly. An informal evaluation takes place with sixty minute-long excerpts used in a variety of styles. The "easy" pieces featuring rock and roll drums "keeping a straight-forward beat" are all tracked successfully. There is a latency between two to eight seconds for the algorithm to begin tracking the signal accurately. Scheirer introduces two important methods into the evaluation process. Firstly, a group of volunteers are asked to tap along to a selection of pieces, thereby enabling comparison of the algorithm with human tapper subjects. Secondly, Scheirer acts as a 'musical expert' tapper, by manually annotating the beat, which provides the 'ground-truth' in the experiment. The algorithm performs consistently as well as the human subjects relative to this 'ground-truth'. The most problematic observation is the algorithm's tendency to drop beats or shift phase.

Klapuri et al. [KEA06] adopt the comb filter method approach of Scheirer and, in order to detect harmonic change in cases where onsets are less apparent, use an increased number of sub-bands. Their method analyses across several different metrical levels. The *tactus* level is the level at which most humans naturally tap along, with one tap per crochet or quarter-note. The level of the *tatum*, a term first used by Bilmes [Bil93] to denote the 'temporal atom' and in honour of Art Tatum, is the lowest metrical level encountered in a piece. Finally, the *measure* level is determined by points of phenomenal accent, such as on the downbeat of every bar. Analysis takes place at the tatum, tactus and measure levels simultaneously.

A weighting system is used to bias the relationship dependencies of simultaneous periods towards the specific integer relationships found in music. Klapuri et al. note that "for example, it happens quite often that one tactus period consists of two, four, or six tatum periods, but multiples five and seven are much less likely in music and thus have lower weights." The distribution is modelled as a Gaussian mixture model to allow some deviation from strictly integral ratios. The Viterbi algorithm

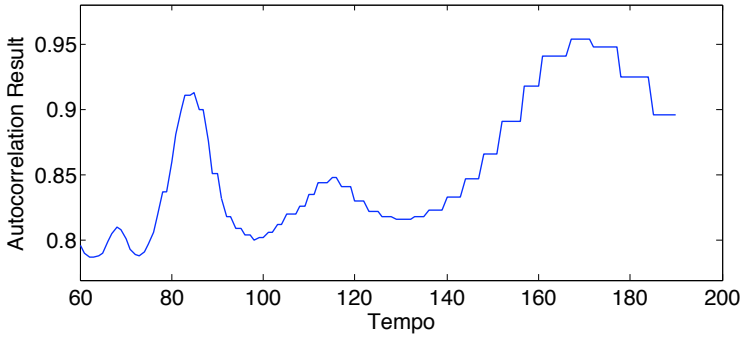

Figure 2.8: Autocorrelation on a stereo recording of drums played by Led Zeppelin's John Bonham. There is a peak corresponding to tempo of the piece at 86 BPM and again at double the tempo.

[Vit67] [For73] is then used to find the optimal sequence of period estimates and phase is calculated after the periods have been decided. The conditional probability between two consecutive phase estimates is modelled as a Gaussian, centered on the previous estimate. Fifteen candidates are generated for both the winning tactus period and the winning measure period. Rhythmic pattern matching aids the estimation of the measure pulse, implying that a model of musical structure informs our perception of measure.

2.4.4 Auto Correlation

Equation 2.4 shows the general equation for an autocorrelation function for time lag τ over N samples. When this method is used on a suitable driving function, such as energy or onset detection, there are generally peaks when τ corresponds to a multiple of the beat period, since musical events happen at those intervals.

$$A[\tau] = \sum_{n=0}^{N-1} x[n]x[n-\tau] \tag{2.4}$$

An example of autocorrelation on a two-minute excerpt of a recording of drums played by Led Zeppelin's John Bonham is shown in figure 2.8. There is a peak at the *tactus* level of approximately 86 BPM and at the *tatum* level of 172 BPM. The magnitude of the difference, however, is

comparatively slight, with the minimum value of the function over 80% of the maximum value. The autocorrelation function can be used to estimate the average tempo for a piece of music, but it does not provide the phase. It has been used in approaches to the problem by Brown [Bro93] to find the meter, and for tempo estimation in Davies and Plumbley [DP07] and Ellis[Ell07].

Ellis uses autocorrelation to provide a global estimate for the tempo, and multiplies the autocorrelation function by a Gaussian weighting window, $W(\tau)$, in order to account for the human tendency to prefer tempos toward 120BPM. Testing this method on the 2004 Audio Contest for Tempo database [GD05], Ellis' algorithm scores 35.7% accuracy relative to an expert, but this rises to 74.4% accuracy if the estimate is allowed to differ by a factor 2 or 3 above or below. Modifications made to the algorithm after testing raised these values to 45.8% and 80.6% respectively.

Davies and Plumbley first calculate a tempo estimation using the autocorrelation function with a comb filter to prioritise tempos in the range 80 BPM to 160 BPM, which correspond to the optimal periods favoured by humans in tapping tasks [DJB00]. Having found a tempo hypothesis, beat alignment is initialised by placing the first beat at the location of a suitable maxima of the onset detection function in the first bar, and placing the next beat by a process of induction. There is a Gaussian weighting bias centered at the beat location predicted by the tempo hypothesis. The standard deviation is set to $\frac{\tau}{4}$, where τ is the beat period, derived to prevent off-beats falling within the window which would result in phase-switching. In order to recover from errors and handle changes in tempo, Davies and Plumbley implement a two-state model, in which the algorithm has a current tempo estimate around which a narrow band of tempos are analysed, and also calculates the best result from a wider range of tempos. If the current estimate fails to be the most prominent tempo for successive results, it changes to the new tempo. Whilst this is a successful strategy for recovery from error, a difficulty can arise when it is used for an accompaniment system if the beat tracker accepts the new estimate too readily during a complex passage, leading to discontinuous tempo estimates. The balance between adaptability and stability proves

consistently hard to define satisfactorily.

Brossier entered a real-time version of Davies and Plumbley's algorithm [DP05] into the 2006 MIREX competition [MMDK07]. An alternative real-time implementation of Davies and Plumbley's algorithm by Stark and Plumbley [SDP08] shows considerable promise as a means for real-time tempo estimation for a wide range of acoustic signals. Nick Collins has developed a a real-time event detection system BBCut [Col05a] which makes use of the autocorrelation beat tracking methods of Davies and Plumbley [DP05] and the onset detection function developed by Bello and co-authors [BDA+05]. By detecting note onsets, this system is suited to performing automated creative tasks based on note onsets within the signal and its low-latency efficiency makes it suitable for the performance of 'live coding' [CMRW03].

2.4.5 Dynamic Programming

Dynamic programming, introduced for beat tracking by Laroche[Lar03], is an algorithmic procedure which recursively defines a score for a path, consisting of a tempo track and downbeat locations, and requires only that this score is a function of the score of the path at the previous frame, the local score of a new candidate and a transition score. In this way, the optimal path can be computed efficiently in linear time.

Ellis [Ell07] constructs a cost function designed to the reward both the strength of the onset at designated beat time t_i and conformity of tempo to the target tempo, determined through the autocorrelation procedure described above:

$$C(\{t_i\}) = \sum_{i=1}^{N} O(t_i) + \alpha \sum_{i=1}^{N} F(t_i - t_{i-1}, \tau_p), \tag{2.5}$$

where $\{t_i\}$ is a set of beat times, $O(t)$ is the strength of the onset envelope at time t, α is a weighting parameter determining the relative importance of onset strength and regularity of tempo, and $F(\Delta t, \tau_p)$ measures the consistency of the inter-beat interval Δt to the target spacing τ_p defined by the target tempo. Dynamic Programming is used to recursively calculate the optimal beat sequence in linear time. The approach is non-causal

since the cost function has access to the strength of the envelope beyond where it chooses to place its beat, and in Ellis' algorithm, the algorithm progresses backwards from the end of the file when computing the path of beats.

2.4.6 Agent Based Approaches

A multi-agent architecture enables several hypotheses to be examined simultaneously. When events have an ambiguous interpretation, this allow the consequences of both interpretations to be evaluated until future observations determine which performs best on the data. Each agent is capable of adapting and evaluating its own behaviour relative to the input and can interact with other agents to perform a given task.

Dannenberg and Mont-Reynaud [DMR87] describe a method for real-time beat tracking that uses a history mechanism, involving a weighted average of previous tempos estimates, in which a decay rate influences the behaviour of the tracker. With the decay close to 100%, the tracker ignores the previous history and is unstable, whilst with a setting close to zero, it is slow to respond. The ability to adjust or automatically select parameters can considerably alter the behaviour of an algorithm, so that for a performance system, the optimal setting can be chosen for the piece. If an algorithm is intended to work across a large database of recorded files, the setting would have to be determined by analysis.

Allen and Dannenberg [AD90] adapted this method to allow the tracker to consider several states, each corresponding to a tempo and phase interpretation, with new events causing each state to generate new states. In this multiple agent-based approach, Allen and Dannenberg limited the number of states by choosing those which had the smallest change of tempo. In order to limit the magnitude of the search, expansion takes place in order of credibility only until new data is received. Also, similar states are merged and states that lack musical credibility are terminated.

Goto and Muraoka [GM94] made the observation that music with drums has a relatively steady tempo, and so the challenge is to locate the beats correctly, rather than focusing on tempo estimation. Their

beat tracking system, BTS, interprets onsets from bass drum and snare drum onsets, although since they deal with stereo files that require pre-processing, Goto and Muraoka report that it is not possible to locate these precisely. They use multiple onset-finders in several frequency ranges which learn the characteristic frequency ranges for the bass and snare drum.

BTS has a bias towards a particular metrical pattern for drums, since bass drum onsets tend to fall on the first and third beats of a bar, and snare drum onsets on the second and fourth. This pattern is a defining characterstic of rock 'n' roll music; the snare on the 'two' and 'four' is called the backbeat [Iye98] and it typifies many of the early record from the fifties and sixties from which other genres of music, such as rock, dance, funk and hip-hop, emerged. In BTS, multiple agents correspond to different strategies for tracking beats, and they make different predictions and evaluate their own reliability. BTS generates information on the basis of the most reliable hypothesis.

Collins's DrumTrack [Col05b] synthesises work by Goto [Got01] and Laroche [Lar03] into a real-time system. Whilst tempo induction is reported to be relatively straightforward, correlation of the energy signal was not sufficient to determine phase alone, and a pattern matching heuristic was used in the manner employed by Goto. Although evaluation shows it does not perform as well as Davies and Plumbley's offline beat tracker, this is to expected since it is tested on material featuring discontinuities in tempo without a prior distribution.

Dixon's BeatRoot algorithm [Dix01] initially has a tempo induction stage, which clusters observed inter-onset intervals to find the best tempo hypothesis, and a subsequent beat tracking stage. Multiple agents are characterised by their state and history, corresponding to a tempo and phase estimate and a history of beat locations. The system is designed to track smooth changes in tempo and small discontinuities. Each agent has an inner window of 40msec from the predicted beat time, within which it will accept deviations, and an outer window of 20% and 40% of the inter-beat interval, respectively before and after the predicted time, representing a change in tempo which an agent accepts as a possibility. If events

are within the inner window, then they are accepted as beat times, with beats in between calculated by interpolation. The tempo is updated by a fraction of the difference between predicted and observed. When an event falls in the outer window, it accepts the event, but also creates an agent that ignores the event, with the choice between them determined by their future scores. Agents which are sufficiently similar result in one of the agents being removed.

2.4.7 Probabilistic Approaches

Hainsworth [Hai03] proposed a particle filtering approach to the problem, based on the application by Cemgil and Kappen [CK03] to beat tracking from MIDI input. He transforms audio using a spectral difference onset detector and the particle filter provides an estimation for the probability of each state given the observation. A Kalman filter is used to update the estimate of the system. One major problem with this model is the computational time required to calculate the probability distribution.

2.5 Discussion

Large and McAuley initially approached the problem from a psychological perspective, investigating the process of entrainment of the listener to a regular beat. Subsequent approaches appear to accept the assumptions implied by this cognitive standpoint, namely that there is a definitive local tempo to find, which gives rise to the observed signal. This assumption seems necessary for the task to be meaningful. However, there are still strong differences with respect to tempo fluctuation between different genres of music. For rock and pop music, the local tempo is often consistent with a global tempo averaged across the piece, whereas in some classical music or, for example, Klezmer music, where some pieces speed up as they progress, only the idea of an underlying local tempo is meaningful, reducing the global tempo to a form of statistical average.

Classical performances are often *expressive*, with soloists or conductors imposing their particular interpretation on a score and the timing of

notes may exhibit strong deviations from strict metrical accuracy. There is considerable emphasis on 'phrasing' where a sequence of notes is grouped together, but transitions between phrases may involve moments of unpredictability, where the listener can no longer gauge the tempo, even at a local level. Such musical forms play with temporal expectation and the cognitive process of entrainment in ways that rock and pop music tend not to. As a result, beat tracking algorithms that aim to work on all genres require a method for recovery from error and more flexibility for tempo change. They therefore compromise their stability by tending to rely on 'the beat' being located at moments when there is a noticeable increase of energy of spectral change in the signal. Likely errors resulting from this approach are during syncopated passages where the stresses are at unexpected metrical locations, potentially causing the beat tracker to change to the off-beat.

With some exceptions, such as Goto and Muroaka's BTS which searches for specific bass and snare drum patterns, a common feature of all the beat tracking systems is their tendency to work as *bottom-up* algorithms, using low level features of the audio to extract tempo hypotheses and then looking within the audio frame for a strong beat location. No *top-down* interpretation guides this process, yet it seems that for humans, our ability to tap to musical pieces is directed by our perception of a higher level metrical and harmonic structure. We may understand the structure from other clues, such as using the pitch of the notes relative to the key to place salient notes or chords in strong metrical positions, in accordance with Temperley's strong-beat rule [Tem99a], but most beat trackers currently do not integrate this kind of harmonic information. Temperley has suggested there is an interaction in our processing of harmonic and metrical information, whereby each forms an input to the other. Cemgil [Cem04] has likened this inter-dependency to a "chicken and egg" problem since "the quantization depends upon the intended tempo interpretation and the tempo interpretation depends upon the quantization". It may be the case that a unified system would perform better than systems with independent rhythmic and harmonic components. Much of the harmonic information is discarded in order to find a suitable driving function for

the beat tracker, such as an onset detection function or Scheirer's [Sch98] sub-band noise. However, if a parallel harmonic analysis took place, that might provide a system with suitable information to locate bar boundaries, detect time signature and distinguish strong from weak metrical locations.

In addition, musicians may use other means, such as visual cues or information gained through rehearsal, in order to synchronise their parts. Desain and Honing [DH99] suggest there may be a strong element of top down processing involved in rhythm tracking. An initial tempo and phase hypothesis may be adopted by humans relatively quickly. Beyond this point, events are interpreted in a top-down manner, in order to guide more precise updates of the hypothesis. Much of the work on beat tracking investigates the bottom-up processing required to make the initial estimate, in which top-down processing has not yet featured.

Where the tempo is relatively steady, less top-down interpretation is required in order to locate beats, and signal processing techniques such as the use of autocorrelation and resonators will pick out regularity of pulse within a detection function. When phrasing or harmonic clues are relied upon by humans, we might expect these methods to experience more difficulty. Davies and Plumbley, Klapuri et al. and Dixon's algorithms all show a wide variation across genre [MMDK07], performing best on world and pop music, which has a strong rhythmic component. This suggests that tacit assumptions of regularity of tempo and the repetitive patterns of strong beats are important factors in their success. Given the inherent stylistic differences with regard to tempo, perhaps particular strategies need to be investigated for different genres, rather than hoping that the algorithm will perform well across a wide database of musical pieces. As Hainsworth [Hai03] points out, "it is unlikely a beat tracker designed for dance music will work on choral music".

Due to local tempo changes, beat positions may vary in definition. Temperley [Tem99a] notes that "the exact location of beats is often somewhat indeterminate". The willingness to test beat trackers on a wide variety of genres indicates an optimism that the same approach could succeed for all. However, often each method has a priori assumptions and

bias concerning the nature of the signal to be tracked. Where this is explicit, such as in Goto and Muraoka's [GM94] restriction to 4/4 meter and inclusion of a pattern bias, the beat tracker is restricted to musical context appropriate to its design.

What unifies the approaches to beat tracking presented here is the assumption of a locally consistent tempo. This can be seen either directly in the use of oscillators or resonators, agents with varying tempo hypotheses or through the use of functions such as autocorrelation, designed to reveal repetition in data. If the local tempo is too variable, none of these approaches would yield a correct result, since beat tracking would require the kind of prior knowledge that musicians use when creating music.

In this thesis, we will only be considering interactive accompaniment for pieces where the tempo is at least locally consistent. Musical forms with a more defined sense of beat tend to adhere to this assumption and explains the increased success of beat trackers on databases of rock and pop music, which tend to have far less tempo variation due to the repetitive percussive nature of the music. As a comparison, Klapuri et al.'s algorithm scores approximately 45% correct on classical, yet close to 90% on rock music.

2.5.1 Multiple Interpretations and Discontinuity

When events are ambiguous, subject to multiple interpretations, and without a score or prior knowledge, one possibility is to adapt the agent-based approach to the real-time scenario, where each agent follows an alternative interpretation of previous events until future information indicates which interpretation should be decided upon. However, a real-time tracker necessarily has to either follow a single course of action, or agent, so there will be inevitable inaccuracies in synchronisation when the wrong agent was chosen. In addition, it raises the problem of finding a suitable musical transition between the multiple hypotheses.

An example of a real-time system which makes use of different modes of behaviour is the two-state model of Davies and Plumbley's system, where one state has a narrow tempo hypothesis, and the other looks analyses

the tempo across the full range of values. This prevents the sudden jumps in period which occurred without narrowing the range of tempos, and by analysing the confidence of the current hypothesis, the beat tracker can resolve itself after errors and not remain fixed at the last tempo. However, a transition to the wider tempo hypothesis still involves a discontinuity in tempo output.

The agent-based approach may be necessary for music with unpredictable timing changes and a wide variety of rhythmic devices. It may be the case that music with strong tempo variations, deviations, syncopation and polyrhythms may benefit from multiple agents acting as insurance against bad interpretation of information. The use of multiple hypotheses may also be crucial to a system's ability to recover from error.

2.5.2 Phase and Synchronisation Accuracy

A common feature of these approaches to beat tracking is their focus upon tempo induction. Clearly this is vital for any offline tracker with no prior information, but often, despite an accurate tempo hypothesis, the trackers will fail to track all the beats in a file. The difficulty in doing so is making the continual adjustments to phase necessary in order to correctly interpret subsequent events and place beats correctly. Although an accurate tempo estimate is a good guide to placing the next beat, any inaccuracy will be amplified over time without re-estimation. Often in the literature, the problem of tempo adaptation and phase adjustment is given less attention than that of tempo induction. However, given the wide range of rhythmic devices that must be correctly interpreted and navigated, it may prove to be as challenging for a beat tracking system as tempo induction and phase determination.

Collins' [Col06] work on autonomous agents for live computer music led to the recognition that "the determination of the phase is perhaps the most critical facility of human beat tracking required for musical interaction". In the case of an interactive system, the phase must not only be continually estimated for the signal, but an appropriate prediction must be made for the future with sufficient accuracy that these predicted beats are

perceptually synchronous with live performers. Without a correct phase estimate, there can be no metrical interpretation of events: a crucial faculty for any real-time system.

In seeking a bound for perceptual synchrony, studies of human behaviour provide boundaries to the limits of acceptability. There is often an asynchrony observed when humans attempt to tap in time with a metronome, whereby subjects tap typically 30 ms *before* the stimulus [Asc02] without consciously perceiving that they are doing so. Whilst subconscious compensation will be made for variations as small as 4ms, Lago and Kon [LK04] argue that synchronisation within the region of 20 to 30ms, equivalent to a distance of approximately ten meters, should be sufficiently accurate so as not to be perceptible. Latencies between 10 and 20ms were not detected at all in tests by Mäki-Patola and Hämäläinen [MPH04], who independently placed the threshold for Just Noticeable Difference (JNS) at 30ms. With respect to a tactus interval, this corresponds to approximately 6% of the beat period and is a reasonable limit for the kind of errors we can expect to tolerate within a real-time system. In performances across a network, it has been observed that latencies greater than 20 or 30ms result in a gradual slowing of tempo [CGLT04]. We shall therefore adopt the 30ms bound as the threshold for perceptual synchrony.

2.6 Languages and Programming Environments

The development of a real-time system requires access to audio inputs from the computer's soundcard and the ability to send audio or MIDI information out of the computer. Most interactive systems therefore make use of existing programming platforms such as Max/MSP, PureData, SuperCollider [McC96], ChucK [WC03] and Csound [VE90] that provide specialised routines to handle audio. SuperCollider is based on object-oriented programming framework, thereby allowing users to create multiple instances of audio unit generators, and is a favorite of proponents of 'Live Coding' for its speed and flexibility.

Max/MSP is a modular graphical programming environment that

emerged from work by Miller Puckette at IRCAM in Paris in the eighties. Whilst IRCAM continued to maintain a version of Max, the Max language was commercialised by Opcode systems and subsequently by David Zicarelli's company Cycling '74 [7]. Whereas the original Max language was designed to handle MIDI information, once processing speeds increased sufficiently, an audio environment, MSP [Zic98], was also developed which can perform DSP operations on streams of audio. Max/MSP thereby allows computer musicians direct access to audio buffers and MIDI information without the difficulties inherent in coding the necessary subroutines at a lower level. Puckette has subsequently released an open source modular environment, Pure Data [Puc96], which is modelled on Max.

Both these environments are designed to operate on input from audio and MIDI sources through a series of linked patches [Puc02], which perform operations in real-time on the streams of audio or numerical data which are the patches' inputs. In response to criticism of the limitations of Max, several authors stress the point that the language also supports the writing of external objects and routines in C. Lippe and Settel [RGD+93] emphasise how this combines the power of programming in C with the fast and convenient framework of Max. The language has recently been extended to include Java and JavaScript. PureData also allows the writing of externals in C and Java. Graphical extensions of the environments, Jitter and gem [Dan97], have been developed which allow the user to manipulate data for visual projections with similar tools.

The Max language itself is limited in its ability to store and manipulate data effectively. IRCAM's FTM library [SBS+05] extends Max by providing the ability to create and manipulate complex data structures. The initial motivation was the requirement for flexible score representations and the need for an efficient representation of matrices and vectors, as used by *Suivi* for the implementation of hidden Markov models used in real-time score following. It is also possible to instantiate data structures within the programming languages that Max supports: C, Java and

[7] `http;//www.cycling74.com` as viewed 7th May 2009

Javascript.

For programming of real-time interactive visuals, OpenFrameworks [8], is a C++ library for visual and creative coding, and vvvv [9], is a programming environment for video which resembles the modular design of Max/MSP. The power of these languages and environments gives rise to interesting new possibilities for musical structures and provides an interface for humans to interact with electronically generated sound.

2.7 Summary

The challenge of developing a system that can maintain a stable tempo and phase hypothesis in real-time is considerable. It is clear from investigating offline algorithms, that we will have to make assumptions about the nature of our signal and use prior information if we are to successfully create a real-time tracking system. Since rock and pop music has a clear rhythmic structure, any errors will be audible to the audience, so we require a system that can reliably track the beats with 100% being longest continuous segment tracked. State-of-the-art beat tracking algorithms attempt to perform the task on all genres of music. We will restrict the genre to the rock and pop music with a strong rhythmic element and provide the beat tracker with individual signals from dedicated microphones rather than a stereo mix. In this way, we hope to develop a method of beat tracking for live performance that is reliable but responsive.

[8] `http://www.openframeworks.cc` as viewed 7th May 2009.
[9] `http://www.vvvv.org` as viewed 7th May 2009.

Chapter 3

B-Keeper: A Real-Time Drum Tracker for Live Performance

We will now present work aimed towards the creation of automatic accompaniment systems for rock and pop music. Since the drums are central to the rhythm of a band, we will focus initially on beat tracking for this instrument.

3.1 Approach

The beat tracking algorithms discussed in Chapter 2 make use of both event-based and signal-based techniques. With the exception of Goto and Muraoka's [GM94] use of templates for rock and pop music, these techniques are intended for a wide variety of musical signals and in the field of music information retrieval, they are often applied to fully mixed stereo files. The goal of these retrieval systems is to enable new functionality for searches on the vast store of music files [Dow08] and the MIREX competition for beat tracking therefore includes music representative of many genres. Thus the beat trackers entered into this competition have to be adaptable for a wide range of audio signals without any prior information.

Here we will make assumptions appropriate for the genre. In rock music and pop, the tempo is often relatively steady across the whole song. The drums are played by striking them with a stick or mallet (an event lasting under 10msec) and so it is relatively simple to extract onsets from the dedicated microphones on the drum kit. We therefore choose to process onset event times rather than use an onset function derived from the audio

signal. We hope that the precision with which these times can be specified will allow us to track the tempo and phase more accurately.

Our approach to the problem makes use of a dual-process mechanism for timekeeper period and phase as observed in literature from music perception [Rep05]. We attempt to directly calculate changes to tempo and phase from incoming events. This requires interpretation of the rhythmic structure and drum pattern, and categorisation of which events for accuracy and metrical position when making updates.

By categorising events relative to a metrical structure, we aim to build a system which can then interpret and analyse other harmonic events relative to their place within the music. This could have significant advantages for the further development of interactive systems. Klapuri et al. [KEA06] and Davies and Plumbley [DP07] have both worked on the extraction of metrical information, such as bar boundaries and on-beat / off-beat classification, but with no prior knowledge of genre or tempo. In our approach, we allow the musician to initialise the accompaniment with an approximation of the correct tempo and phase, then use the metrical structure imposed by the audio sequencer to maintain the correctness of the rhythmic alignment. This provision of prior knowledge is justified by appeal to performances with human musicians where a band knows the approximate tempo, either learned from rehearsals or through a count-in. In this way, we avoid the need for tempo, phase and bar boundary estimation which remains a problem for future research.

Event-based approaches to accompaniment have been used by Large [Lar95] and Toiviainen [Toi98], who used them to update the frequency of oscillators. Another parallel with the oscillator model is the integration of expectation into the algorithm. As an event's location relative to the period and phase of the oscillator determines the update, we will search for new beats within a window around the expected beat locations, employing an explicit rule-based (using top-down processing) reasoning, dependent upon factors such as beat location and recent history.

Initially, we designed the drum tracking algorithm to process only audio from the kick drum since this loud event seems to define the beat [Iye98], but once we began testing the system, added the signal from the

microphone on the snare drum. In the development of B-Keeper, we assume that all onset events are intended to occur on eighth-note divisions of the bar, without any expressive timing deviation. By use of a weighting system, we favour the more important beats, the kick drum on the 'one', and the snare on the 'two' and 'four', with syncopated beats less likely to cause phase-correction. We adopt the proposition from the GTTM and established theories of meter that a hierarchy of beat locations exists and, accordingly, we will react to an event depending on the beat's location within the hierarchy, our recently observed history and our confidence level in the system's predictions. In Chapter 5, we shall introduce a fully-layered hierarchical system, which confidently ignores events at lesser beat locations on the condition that it has observed recent regular events at a higher level.

Despite the potential existence of small timing deviations, we will design our beat-tracker to act as if all beats are placed to a metric grid, so that a snare at a strong metrical location that is marginally late will cause a small phase-correction towards the new event. If we consider again, Figure 2.2, discussed in the previous chapter, then a choice can be made in light of our discussion on the style of drum playing we may encounter. Our salient events, the kick drum and the snare backbeat, will be loud acoustic events at strong metrical positions of the bar. It makes sense to assume that these *define* the beat. This is equivalent to stating that the main drum events *do not* exhibit expressive timing.

Examining John Bonham's drum files, although there is evidence that certain drum hits are deliberately displaced and played late against the beat, we could equally regard these as exhibiting local tempo change, whereby the event and future events are displaced but the tempo remains constant. Under this interpretation, if an expressively timed event, such as (a) in figure 2.2, was interpreted as a local tempo change, (b), then the next onset event would be interpreted as a local tempo change in the other direction. However, our system would have *reacted* to the beat, and in the event that it constituted an even greater change, a global tempo change, (d), then at least it has reacted towards accommodating this global change. At present, the danger of a lack of response through disregarding actual

local tempo change outweighs the benefits of smoothness or sensitivity to expressively timed events.

In rock music, despite the evidence that players sit forwards or backwards relative to the beat, the argument that salient events in the drums *define* the beat and are therefore local tempo changes seems to be consistent with their role musically in a band. The essential difference between error and expressive timing is that the latter is a regular rhythmic aberration. If, in future, our system could record the tendency to displace beats with expressive timing, then we could accommodate this behaviour, but to incorporate it into the initial design might cause the system to be unresponsive. We will describe further developments making use of top-down metrical structure which allow for some expressive timing in Chapter 5. We therefore aim for our metronomic click to sit at the same place, relative to the perceptual beat, as the drum events we process from the microphone signal.

Whilst multiple agent hypotheses might enable versatility to negotiate tempo variation and recover from error, we chose to investigate what can be achieved with a single hypothesis on the basis that since a human can successfully tap to rock drums, then in principle a single tempo and phase hypothesis can be maintained. This means that we are restricting the range of our beat tracker to more conventional rhythmic patterns, in order to gain a stability for the resulting system. We hope to recover from errors by changing modes of behaviour and parameters, rather than switch between discrete hypotheses.

3.1.1 Model Assumptions

The aim of the drum tracker is to detect the underlying metrical pulse which explains the occurrence of onsets. We would like this to be stable despite timing discrepancies between notes and changes in the rhythmic pattern of the signal. To this end we make some initial assumptions:

- The audio signal is rhythmic and is underpinned by regular beat to which a human could be entrained.
- An approximation of the tempo is already known by the system.

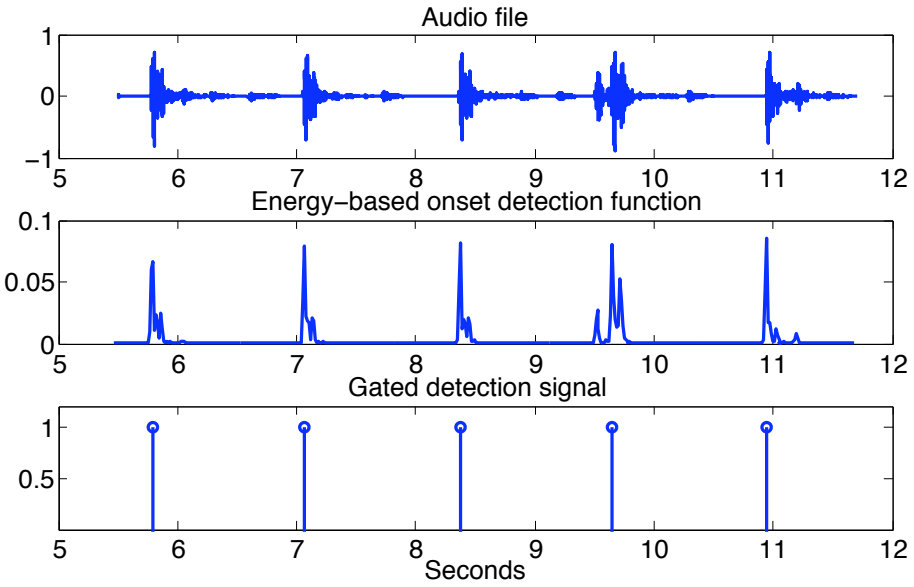

Figure 3.1: Signal from bass drum microphone and the pre-processed signal using an energy-based detection function.

- The tempo is relatively steady, with only small fluctuations occurring.
- Onsets occur on a metrical level that are even subdivisions of the beat.
- Variations between the inter onset intervals indicate some fluctuation in tempo.
- We assume that events are not timed expressively, but are displaced either as a result of local and global tempo change or performance error.

3.2 Implementation

During development of the algorithm, we used recordings of the bass drum microphone of a live drum kit being played in a room with a full band. There is a comparatively low amount of noise from other percussive or rhythmic instruments other than the bass drum. The signal and the output of an energy-based onset detector can be seen in figure 3.1.

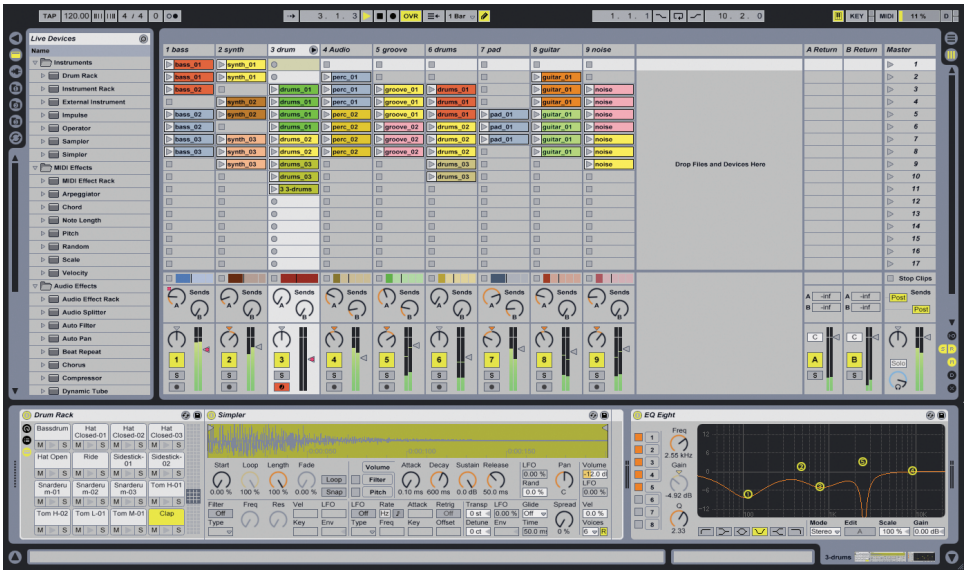

Figure 3.2: Screenshot of Ableton Live's Session View.

Initial development of the algorithm was carried out offline in Matlab using an energy-based onset detection function from Bello et al. [BDA+05]. Subsequently, the algorithm was coded as a Java external within Max/MSP, with an interface designed to allow the user access to control parameters which affect the behaviour of the beat tracker. This implementation made use of Miller Puckette's *bonk~* [PAZ98] object for Max/MSP, which is suited to the detection of percussive onsets. The *bonk~* algorithm makes use of spectral change and has a low hop-size of 256 frames (5.8msec at 44.1kHz sampling frequency). The supervisor sets an appropriate threshold for the onset detector via the B-Keeper user interface in Max. In order to provide real-time automatic accompaniment, we made use of Ableton Live [1], a popular audio sequencer with D.J.s. which, due to its time-stretching features, whereby the tempo of an audio excerpt is changed, is highly suited for our purposes here. This software has a professional user interface, familiar to musicians, that can be controlled externally through the sending of MIDI information. The session view of Ableton Live is shown in Figure 3.2.

Offline tests demonstrated that the system could fall towards local

[1] `http://www.ableton.com` as viewed 14th April 2009

attractors over the tempo range. These are corresponding maxima for autocorrelation-based processes, such as seen in Figure 2.8 in Chapter 2, where 85 BPM, 170 BPM and 117 BPM are all local maxima. When determining tempo estimates from inter-onset intervals (IOI's), if a regular pulse is tracked at a tempo whose ratio with the original can be expressed as fraction with integer components, then the tracking tempo will interpret some intervals as integer IOI's at the tracking tempo. For instance, if the tracking tempo is four-thirds the original, then every third beat of the original will fall on the fourth beat of the related tempo. Hence, one must either have a good local approximation to the tempo (the approach taken here), or examine tempo hypotheses over a wide range of tempos, such as in the beat tracking methods discussed in the previous chapter.

3.3 Algorithm Description

For a given tempo, the *tatum*, the name given to the temporal 'atom' by Bilmes [Bil93], is the duration of the high-frequency pulse at the lowest metrical level. Here, we shall take it to be the duration of an eighth note, measured in milliseconds, although technically it could refer to a subdivision at an even lower metrical level. The algorithm takes as input the onset time t_n, determined by the CPU of the computer when the n^{th} onset has been detected. The problem is formulated as how to best align a sequence of drum event onsets, where the n^{th} onset occurs at time t_n, with a regular click track at the tatum level from a sequencer at times x_n or $E[t_n]$. The onsets are presently assumed to be intended to occur on the beat, whilst the tempo is steady but not necessarily constant.

We aim to minimize the error between the two sequences by changing the tempo of the click track on the basis of current observed onsets. Early investigations of our beat tracking system confirmed the need for a specialised phase-correction stage as exists in some psychological models [WK73] [DW96]. Our approach to beat tracking uses two dedicated processes: one controlling the underlying tempo and a phase-correction process which quickly adapts to new information. Two processes work in parallel: one to adjust the general tempo and one to synchronise the phase

of the tempo. These are referred to as tempo tracking and synchronisation respectively.

For each onset, we calculate the corresponding inter onset intervals between our new event for all recent onsets. A decision mechanism, described below, weights these results and adapts the tempo estimate accordingly. In addition, a parallel process makes adjustments required for precise synchronisation, equivalent to matching the phase of the sequencer with the live drummer. The synchronisation process makes temporary adjustments to the tempo of the accompaniment to account for phase differences and accommodate local timing deviations, whilst the tempo process adjusts the underlying global tempo. In practice, the algorithm synchronises first and then calculates the tempo, since the synchronisation process can inform the tempo algorithm which beat of the bar the new onset falls on. However, for the purposes of understanding how the two processes work together, we shall describe the general tempo adjustment prior to the phase adjustment.

3.3.1 Tempo Tracking

Given a recent onset at time t_n, we calculate all recent inter-onset intervals between this onset and recent onsets, t_{n-k} and their interpretation in terms of the current tempo hypothesis.

$$I(k) = t_n - t_{n-k} \tag{3.1}$$

$$v(k) = \text{round}(\frac{I(k)}{\tau}) \tag{3.2}$$

The duration, $I(k)$, corresponds to $v(k)$ tatum intervals, where τ is the tatum in msec, which has been defined here as the duration of an eighth-note.

The error between the observation and the value for the inter-onset interval (IOI) predicted by the onsets t_n and t_{n-k} is:

$$\epsilon_{n,k} = I(k) - v(k).\tau \tag{3.3}$$

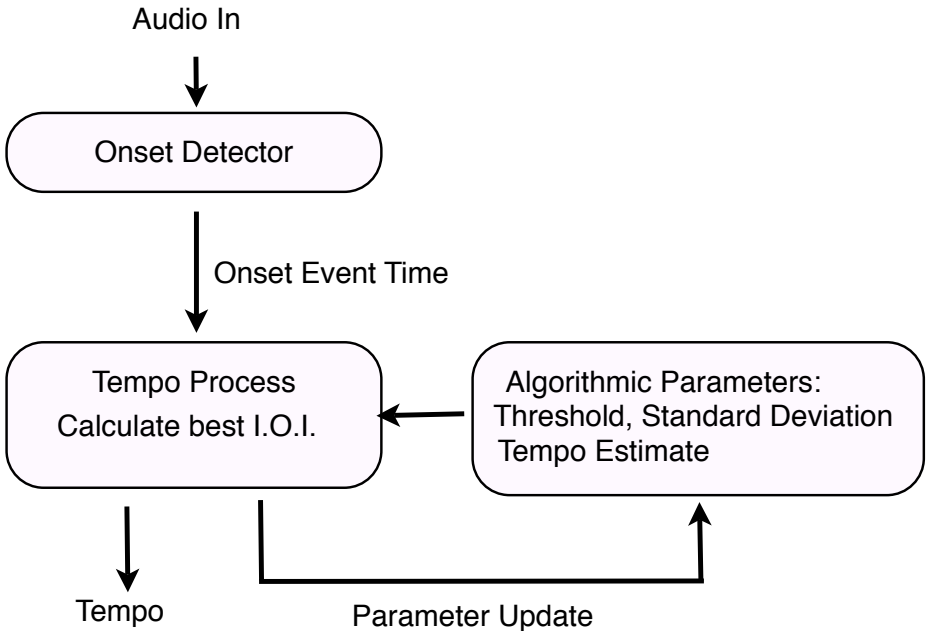

Figure 3.3: Diagram showing the basic structure for the Tempo Process.

The likelihood for an IOI of j tatums occurring is set by the user as $L_{tempo}(j)$, where $1 \leq j \leq 16$, since intervals greater than 16 tatum lengths, or two bars, will not influence the tempo tracking algorithm. This likelihood function is set prior to the piece. Since we expect inter-onset intervals of a beat, two beats and a bar long, we set the corresponding values $L_{tempo}(2)$, $L_{tempo}(4)$ and $L_{tempo}(8)$ close to 1, and less likely values such as $L_{tempo}(5)$ are close to, if not, zero. In general, a 4/4 beat would suggest having non-zero values for powers of two only as the rhythm is cyclical for a regular a power of two (typically 8). Whilst syncopation dictates that other intervals will be observed, there will still consistently be intervals which are a power of two that can provide a stable way to gauge the tempo.

Both the tempo tracking and the synchronisation functions make use of a Gaussian window around the corresponding error terms, so that onsets indicating a small tempo fluctuation are used by the system to adjust its tempo estimate, whereas more radical changes are interpreted as performance errors. The accuracy determined by onsets t_n and t_{n-k}, relative

k	$\epsilon_{n,k} = t_n - t_k$	$v(k)$	$g(\epsilon_{n,k})$	$L_{tempo}(v(k))$	Acc(k)
1	-1.6	2	0.996	1	0.996
2	-8.2	3	0.901	0	0.0
3	-14.8	4	0.712	1	0.712
4	-4.7	6	0.966	0	0.0
5	-6.3	8	0.940	0.92	0.864
6	-7.9	10	0.907	0	0.0
7	2.1	12	0.993	0.68	0.675
8	0.5	14	0.999	0	0.0
9	10.6	16	0.840	0.8	0.672

Table 3.1: A recorded example of the list of recent onsets and the corresponding evaluation for the tempo tracking process. The winning onset is $k = 1$.

to current tatum estimate τ is given by:

$$\text{Acc}(n, k, \tau) = g(\epsilon_{n,k}).L_{tempo}(v(k)) \tag{3.4}$$

where

$$g(\epsilon) = e^{-\frac{\epsilon^2}{2\sigma^2_{tempo}}} \tag{3.5}$$

The Gaussian function, $g(\epsilon)$, given by equation 3.5, has been adapted from the normal Gaussian, or bell-shaped curve, by omitting the scaling factor required by probability theory so that the integral across the reals is equal to 1. Instead, the function $g(\epsilon)$ has a maximum value of 1 when there is no error, and the function decreases as the error increases according to the standard deviation σ_{tempo}, a parameter of the tempo tracking process.

We evaluate $\text{Acc}(n, k, \tau)$ for all recent onsets that occurred during the last two bars. This creates a list, as seen in table 3.1, evaluating the interval between the current onset and all recent onsets with respect to the tempo hypothesis.

Then the greatest accuracy value is given by the inter-onset interval between t_n and $t_{k_{win}}$. We make use of the following update rule:

If $\text{Acc}(n, k_{win}, \tau) \geq \theta_{tempo}$, then $\tau = \tau + \Delta\tau_{tempo}$,
where

$$\Delta\tau_{tempo} = \alpha.g(\epsilon_{n,k_{win}}).L_{tempo}(v(k_{win})).\frac{\epsilon_{n,k_{win}}}{v(k)}, \tag{3.6}$$

and $\theta_{tempo} \in [0, 1]$ is the update threshold. The change in tempo resulting from equation 3.6 uses the inter-onset interval which is in closest agreement to our current estimate, whilst factoring in our prior likelihood of our observing such as interval within a drum signal. This method rejects the less reliable observations (those falling outside the window) and calculates an average of the tempo based only on recent IOI's which fall within the window. In their an agent-based approach, Allen and Dannenberg [AD90] limited the number of states by choosing those which had the smallest change of tempo. We have taken a similar approach here when adjusting the tempo, favouring *tempo coherence*, over all other interpretations. An alternative is to take a weighted average of those inter-onset intervals above a suitable threshold.

3.3.2 Automatic Tempo Parameter Adjustment

The parameters used in the tempo process are the update threshold, θ_{tempo}, the standard deviation, σ_{tempo} (given in msec), used in the Gaussian function which determines the width of the window, and the likelihood of observing individual inter-onset intervals, set by the weighting function $L_{tempo}(v(k))$. In order to find the optimal values for the style of playing, an additional feature automatically adjusts the threshold and standard deviation parameters dynamically.

If the user sets the threshold and standard deviation, then there is a danger that if the window is too narrow, the algorithm may fail to respond to tempo change since IOI's will fall outside the window, whereas if the window is too wide, the system may respond to inaccurate onsets or misinterpret events, such as syncopation. In addition, drum fills present a problem as they generate a succession of rapid IOIs which are to be averaged over. Considerations which account for drum fills will be presented in Chapter 5 of the thesis.

By adapting the parameters automatically, if the tempo is relatively steady, then the threshold will rise and the standard deviation lessen, so that an onset that is then less accurate creates IOIs that fall outside the window and will not affect the tempo. This prevents the system

becoming overly responsive. If successive onsets fall outside the window, the parameters will adjust by lowering the threshold, so that the new IOIs are accounted for. This results in a slower response to tempo change but significantly greater reliability since the algorithm naturally finds settings which suit the accuracy of the majority of onsets detected.

We make use of the following rule to update our parameters θ_{tempo} and the window size σ_{tempo}:

If $\text{Acc}(n, k_{winning}, \tau) \geq (\theta_{tempo} + 0.1)$, then
$\theta_{tempo} = \theta_{tempo} + 0.3.(\text{Acc}(n, k_{winning}, \tau) - \theta_{tempo} - 0.1)$.

Otherwise, if $\text{Acc}(n, k_{winning}, \tau) \leq \theta_{tempo}$, then $\theta_{tempo} = 0.6\theta_{tempo}$.

The window adjusts itself so that the median result of the Accuracy function, described in equation 3.4, pivots around the value 0.7, an arbitrary value chosen so that the average accuracy weighting for onset intervals is reasonably high. Since the accuracy will be in the range $[0, 1]$, our choice of value reflects the fact that we wish half the onsets to result in an accuracy of 0.7 or more. By using an equilibrium point, we ensure that the system adjusts its parameters to respond well when set to automatic mode since the accuracy result also determines the proportion of synchronisation. Hence, if the accuracy is less than 0.7, it widens the window, whilst if it exceeds 0.7 then it narrows the window:

$$\sigma_{tempo} = \sigma_{tempo}.(1 + [0.7.L_{tempo}(k_{winning}) - \text{Acc}(n, k_{winning}, \tau)]) \quad (3.7)$$

3.3.3 Synchronisation

As well as tracking changes in tempo, it is important to also make adjustments to preserve an accuracy in the phase of the estimate or else there will be a drift in alignment between the accompaniment and the live drums. A similar strategy is used to synchronise onsets t_n to the corresponding click track event at time $E[t_n]$. The event is categorised by calculating the accuracy measure described below for the recent and predicted click times, and choosing that for which the result is higher. It is not necessarily the

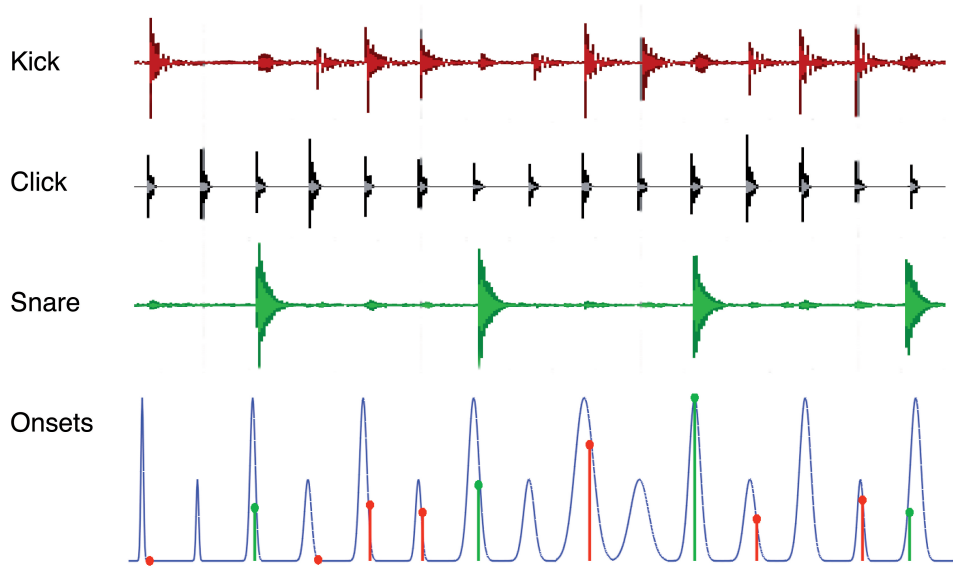

Figure 3.4: Illustration of the how the synchronisation process for kick (red) and snare (green). The accuracy values of onsets result in automatic adjustment of the width of the Gaussian windows around the expected beat locations to maintain synchronisation.

closest click event due to the weighting function employed.

The error between the expected onset and the observed onset time is:

$$\epsilon_n = t_n - E[t_n] \tag{3.8}$$

In performance, we expect onsets to exhibit inaccuracies due to human imprecision, as well as *expressive timing*, where the discrepancy between where the onset and expected beat time is deliberate. As stated in our assumptions, we shall make no distinction between the two during the synchronisation process. In order to do this, we make use of a similar accuracy function to that employed in the tempo tracker. Given an onset at time, t_n, we estimate its accuracy relative to our tempo hypothesis as:

$$\text{Acc}(t_n, \tau) = g(t_n - E[t_n]), \tag{3.9}$$

where the function $g(\epsilon)$ is the same as used in equation 3.5, except the standard deviation is now σ_{sync}, a specialised parameter for the synchronisation process. We also make use of a weighting measure, $L_{sync}(k)$, which corresponds to the likelihood that an onset occurs

at beat k of the bar. The measure $L_{sync}(k)$ functions as the drum pattern that we expect to be synchronising to and can be set by the user prior to playing. Typically, one expects kick drum beats to fall on the 'one' and the 'three' of a bar, not one eighth-note in. Hence, one might choose to set $L_{sync}(0)$ to 1 and $L_{sync}(1)$ close to 0. In practice, we wish to favour onsets on the beat and off-beat, so have used the setting $L(2k) = 1$, $L(2k+1) = 0.4$ $(0 \leq k \leq 7)$, widely in development.

The updated accuracy function becomes:

$$\text{Acc}(t_n, \tau) = g(t_n - E[t_n]).L_{sync}(n_{bar}) \tag{3.10}$$

where n_{bar} is the position of the onset within the bar in terms of tatum lengths from the *one*. We synchronise with the beat if $Acc(t_n, \tau) > \theta_{sync}$, in which case we add a synchronisation factor to our tempo estimate:

$$\Delta\tau_{sync} = (\frac{g(t_n - E[t_n]) + \beta}{\beta + 1}).g(t_n - E[t_n]).L_{sync}(n_{bar}).(t_n - E[t_n]) \tag{3.11}$$

where $0 \leq \beta \leq 1$ is a user-defined parameter which affects the extent to which the system makes the corresponding phase adjustment for observations away from the expected beat location. By setting β close to 1, the value of the first fraction is increased for lower values of $g(t_n - E[t_n])$ so that phase synchronisation for all observations over our threshold.

3.3.4 Decision Tree

Although the metrical structure can aid the calculation process, as discussed previously, events do not always have a univocal interpretation. By adjusting the parameters used in the decision making process, we can change the behaviour of the algorithm as well as the output. In particular, by widening the window and lowering the threshold, we can follow a compromise between interpretations of events, so the model is more willing to accept future information from whichever hypothesis turns out to be true.

In the case of synchronisation or phase-correction, we employ a decision-making mechanism to decide between local tempo change, which we will correct for, and performance error or events to be discounted.

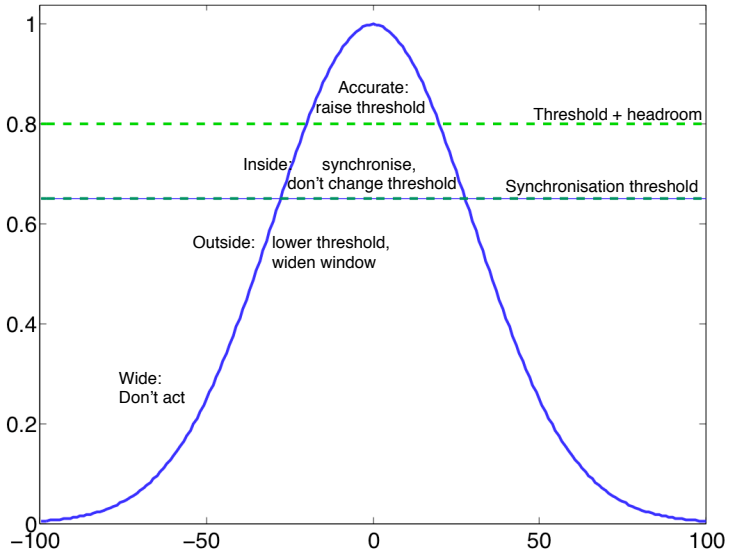

Figure 3.5: Illustration of the different regions for decisions taken by the synchronisation algorithm.

There are three zones into which the beat can fall. These can be seen in figure 3.5. If the expected location of the onset is very accurate, above the threshold plus the headroom, then we synchronise and alter the parameters so the threshold increases and the window narrows. When the onset is above the threshold but within the headroom above it, we synchronise but do not change the model's behaviour. If the onset is below the threshold, then the closer it is to the threshold, the greater the adaptive response of system parameters by the algorithm, whereby the window widens and the threshold decreases. The effect of these three zones on the window size and threshold can be seen in figure 3.7. The plateau corresponds to the area where onsets result in phase-correction through synchronisation, but no alteration of the model's parameters.

This synchronisation factor is added over a short number of intervals, l_{sync}, set by the user so as to smooth the effect of resynchronisation. Hence, our tempo adjustment is given by:

$$\tau = \tau + \Delta\tau_{tempo} + \frac{\Delta\tau_{sync}}{l_{sync}} \tag{3.12}$$

until the synchronisation has been achieved and $\Delta\tau_{sync}$ is reset to zero.

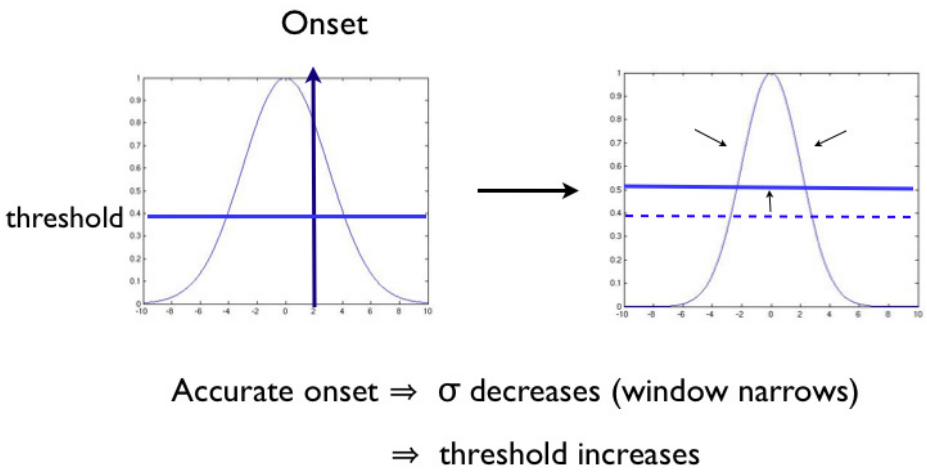

Figure 3.6: Automatic adjustment of synchronisation parameters. The accurate onset results in the threshold increasing and a decrease in the standard deviation used.

3.3.5 Automatic Synchronisation Parameter Adjustment

A corresponding automatic adjustment feature is included in the synchronisation algorithm as well as the tempo tracking algorithm. If the synchronisation error is so large that the onset is outside the Gaussian window, then for future onsets, the standard deviation is increased so that the window widens and the threshold is also decreased. This has the effect of adjusting the parameters to suit the performance style of the drummer and to dynamically change these variables within a performance for optimal response from the system.

Figure 3.7 shows the resulting threshold for different accuracy and beat probabilities. The threshold increases when accurate events are observed at likely locations, and decreases most significantly when events are observed just below the threshold. This can be seen as sharp shelf at the current threshold of 0.6.

3.3.6 Drum Pattern Recording

In order to aid the tracker, we record the pattern of drum events that has been observed and use this within the weighting process. The pattern is non-zero at the corresponding beat location for each observed onset and

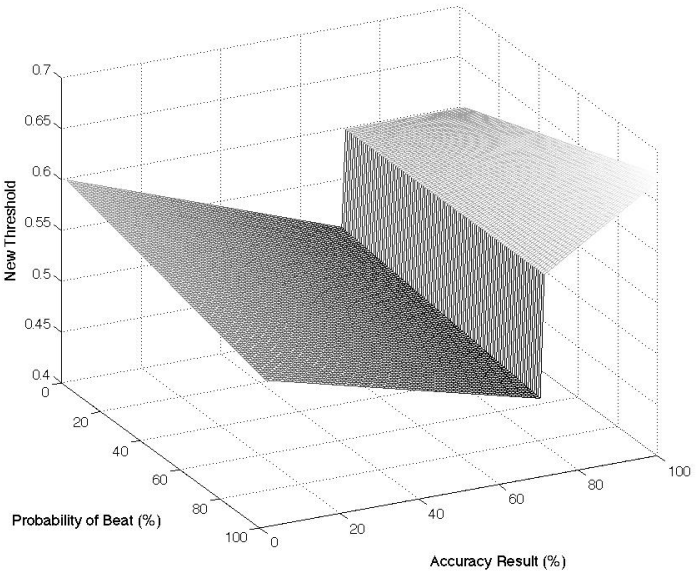

Figure 3.7: Projected threshold adjustment as a function of beat probability and Gaussian accuracy result. The original threshold is 0.6 and the expander and contractor parameters are 0.3 and 0.25 respectively.

its value is given by the Gaussian function of the accuracy measured prior to any weighting. The drum pattern is therefore a direct indication of the expected accuracy of a beat at that location in the bar. In later versions of the tracker, rather than employing a fixed weighting, we used a linear combination of the fixed prior weights and the observed weighting during the performance. This allows the tracker to adapt to a syncopated rhythm and 'learn the beat'.

3.3.7 Supervisor Controlled Functions

The algorithm described above makes use of the following parameters which control the behaviour of the system and are adjustable by the user or a supervisor via an interface in Max. An early version of this interface is shown in Figure 3.8:

- The initial starting tempo should be set by the user. There is also a function for the drummer to set the tempo by playing a set number of evenly spaced beats of the kick drum (or clicking the sticks on the

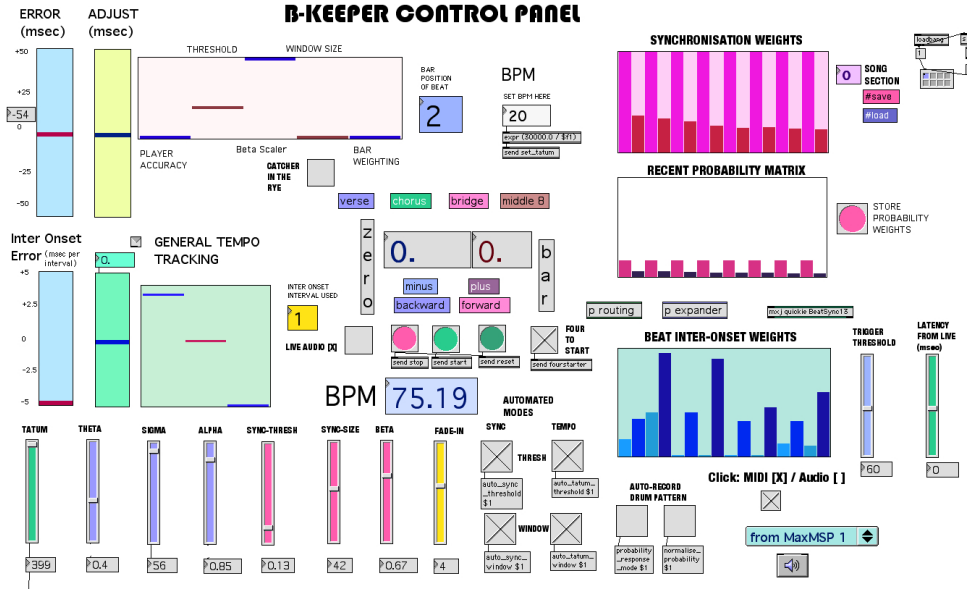

Figure 3.8: Early version of the user Interface in Max/MSP, allowing the manual setting of initial parameters and weights, and access to rescue functions.

snare microphone), where the average interval is used to determine the bpm and send a starting signal.

- α : the scaling factor in the tempo tracking process affects how responsive the system is to changes in tempo. Experimentation has suggested that values between 0.3 and 0.6 give good results. Although it is possible to adjust to every change in tempo indicated by incoming onsets, in practice, it is better to find a balance between stability (α close to zero) and responsiveness (α close to one).

- θ_{tempo} and σ_{tempo} are used by the tempo tracking algorithm to decide whether the observed inter-onset interval will be used to update the tempo estimate. The parameter σ_{tempo} determines the size of the window of the Gaussian function, so if σ_{tempo} is large or the threshold θ_{tempo} is low, then inter-onset intervals are more likely have accuracy functions that exceed the threshold, θ_{tempo}, and the system is more responsive to fluctuations in tempo.

- The parameters θ_{sync} and σ_{sync} determine the threshold and the window size around the expected beat locations for the synchronisation

process, determining in a similar manner how responsive the system is to local tempo change and adjustment of phase.

- β is a scaling factor, analogous to the parameter α in the tempo tracking process, which affects the extent to which the system synchronises in phase to an onset.

- Automatic modes are provided for the θ_{tempo}, σ_{tempo}, θ_{sync} and σ_{sync} parameters in which they are adjusted to match the playing style of the drummer. An optimal balance is sought, where beats regularly result in adjustment of tempo. The algorithm describing this is in equation 3.7.

- The system records where onsets occurred for both the kick drum and the snare drum in the recent playing history. It is possible to use a linear combination between the fixed weights and the weighting determined by this data in the synchronisation process, allowing the system to 'learn' the current drum pattern and use the resulting weights when synchronising with the player.

- Nudge functions have been included which enable the supervisor to rescue the system from misalignment. If parameters are not set optimally or the performance in some way throws the system, the supervisor can see that it may be a tatum interval or a beat out in its synchronisation. Sometimes a drummer may compensate for this and the system will synchronise again, but the *forward* and *backward* function, speed and slow the backing respectively via intervention from the supervisor. A *bar* function also informs the system when pressed, that that location is the beginning of the bar and tempo adjustment is made to realign.

- It is possible that the internal beat position used by the algorithm might differ from the correct metrical position for the accompaniment part. For instance, the system might interpret an onset as occurring on the four tatum intervals into the bar when in fact it is two intervals in. In this case, the misalignment is not audible, but if different parts and audio effects are cued relative to the song position, it is

important to rectify. The *plus* and *minus* functions, simply add and subtract to the beat position to correct such an error. Provision has also been made to ensure that the internal beat position always matches that of the sequencer by means of a MIDI messaging.

- Drum patterns can be loaded as pre-stored weight settings for the synchronisation process. We have made use of the FTM library [SBS+05] to store the matrix using the SDIF [WCF+99] format. It is then possible to automate changes between verse, chorus and bridge in which the synchronisation weights change for each section.
- The *latency* between the audio inputs and the click track can be set by experimentation on the user's part. However, a strategy to avoid latency is described below.

3.3.8 Real-time constraints

Since the algorithm is operating in real-time, some scheduling issues arose during the implementation. Max/MSP's scheduler gives priority to the signal processing stream over the event-based stream since interruption to audio would be more problematic in general than small amounts of latency in numerical processes. However, events are time-stamped so that when an onset event is processed by the B-Keeper system, it has an associated cpu-time from the onset detector *bonk*~.

As the system increased in complexity, very occasionally we experienced time-stamped events being received in a non-linear order with respect to the time they occurred. So, for instance, it is possible that an onset at time t_n is reported later than a click track event which has a time-stamp greater than t_n. As a result, the coding includes sub-routines to ensure that the recent and predicted onset times based on the click track information are in accordance to the cpu-times of the audio onsets.

One potential problem is latency from the drum microphone inputs which have to pass through the analogue-to-digital converters of the soundcard and then the onset detector. The latency of the onset detector is 256 samples (5.6 msec) and the buffer of the soundcard is typically in the region of 256 to 512 samples (5.6 to 11.2 msec). Hence, there is a latency of

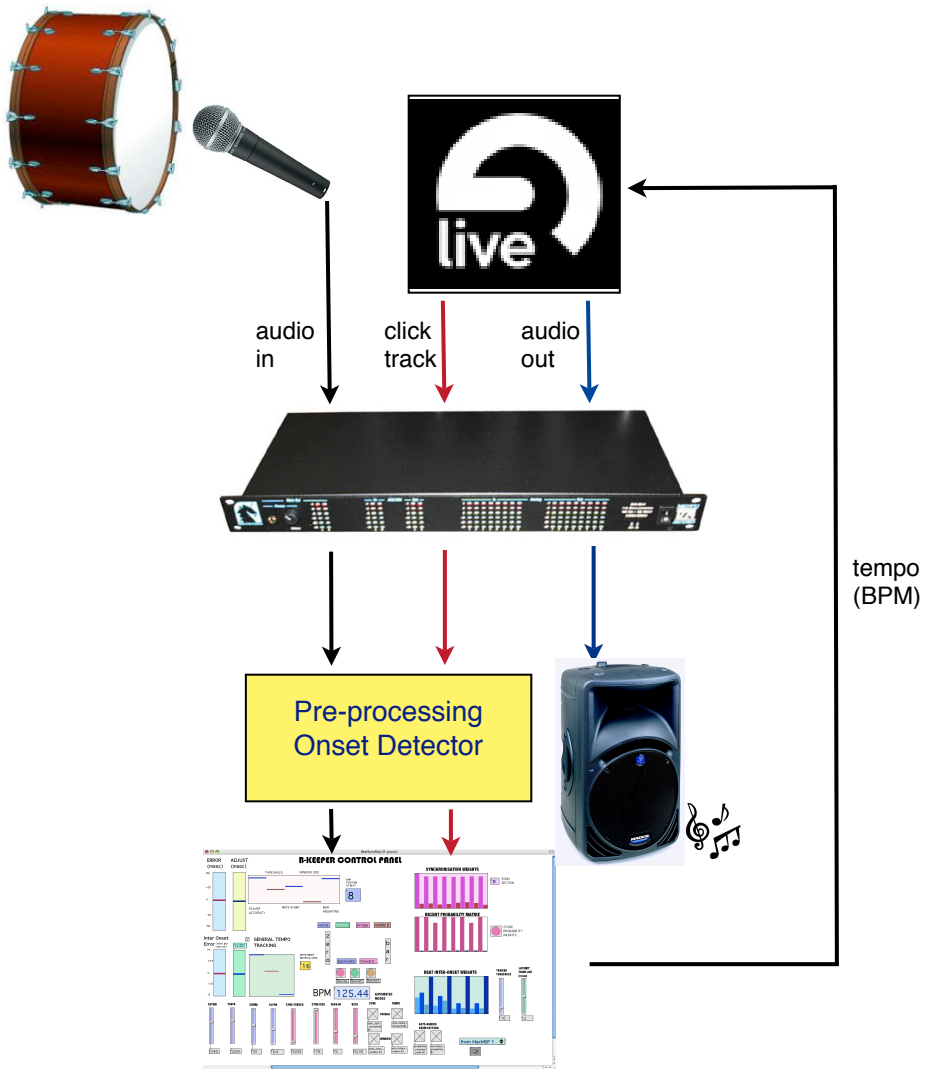

Figure 3.9: System set-up to cancel latency

approximately 10 to 20 msec between the percussive attack and the onset being recorded in the system. The click track was initially implemented as a MIDI message sent from Ableton Live (or from an FTM MIDI track if one is synchronising within Max/MSP) and the latency for MIDI is only one or two milliseconds. It is known that latency between performers of more than 30 milliseconds causes a progressive slowing down [CZS+05]. In this case, although the tempo tracking algorithm functions independently of the click track, the synchronisation algorithm is directly affected by any

latency. This could result in a performer constantly adjusting to synchronise with an accompanying audio track that is continually slipping out of time.

To negotiate this problem, the best solution seemed to be to use the same kind of signal path for both audio and click track inputs. Hence, as well as sending audio accompaniment, we send an audio click track (a regular sequence of percussive sounds aligned with the tatum interval divisions of the bar) out of a spare soundcard output. This is then passed directly back into the soundcard (hence is subject to the same analogue-to-digital conversion process) and through the *bonk*~ onset detector in Max/MSP as shown in figure 3.9. Therefore, using this method, the latency is the same for both the audio input from the drums and the click track from the sequencer.

3.4 Reports from Initial Trials and Performances

Some qualitative feedback was provided at an early stage in the algorithm's development. These reports helped to inform the direction for research and indicate where improvements could be made. An early performance with the system have been given at the Live Algorithms for Music (LAM) Conference at Goldsmiths College, London, December 2006. This performance was with an earlier prototype of the system that did not feature the modes to tune parameters to the drummer automatically. Instead, the threshold and standard deviation parameters were set by the supervisor (AR), and it required manual adjustment during the concert to find the right balance for a stable but responsive system. For this, the threshold and windows were adjusted and the nudge mechanism was used to change the internal metrical position of the drum tracker. A video excerpt of the performance is visible on the internet [2].

David Nock, audio engineer and drummer, who performed at LAM, commented on the experience: *"It has elements of a human - almost an early learning one - you have to guide them through, accommodate them*

[2] `http;//www.youtube.com/bkeepersystem/`

a little." He described the sense of interaction as *"very good. Once it gets in sync, you can feel where its boundaries are. It's able to accelerate and decelerate. Like with a real player, it's a two way process."*

This certainly indicated potential for the method, however, the difficulties in setting parameters experienced in the concert suggested that these should be automated. Further tests on the system were carried out in May 2007 with Joe Caddy, session drummer with 'Captive State', and Rocco Webb, session drummer with Cerys Matthews. The addition of automatic modes significantly improved the system's ability to home in on a regular beat and adapt to find settings suitable to the playing style. The tracker was demonstrated at the Centre for Digital Music Summer Concert at Queen Mary University of London, June 2007, and at the DMRN 2007 Conference at Leeds University.

Joe Caddy described the current limitations of the system:

> "*Some of that was great. There are problems with immediate tempo changes, where if that was nailed, it would make it a great system, wouldn't it? There are some times where, I don't know, maybe it's super-intelligent. It sort of locks into a tempo and if you try to pull away from it, it almost wants to bring you back.*
>
> *I tried changing some of the bass drum patterns and it dealt with that. The main thing is it needs to be much more sensitive. When you start to play something a little more intricate or start to change tempo a little bit, it doesn't recognise it until one or two bars. A couple of times it didn't, it was too slow. You want the recovery to be milliseconds."*

The perception of a resistance to change is an interesting point here. Although Joe would like more responsiveness, the tension between accompaniment and drummer is preserved with the accompaniment resisting change. A drummer has more freedom for expressive patterns against a more steady accompaniment than they would against an accompaniment that instantly adapted. By not using rival tempo and phase hypotheses, the system doesn't make sudden jumps to other tempi as occur when using other real-time beat trackers. This is potentially a disadvantage in

situations where sudden tempo changes might occur, such as free improvisation, but when utilised on steady tempo music, the advantage is a more stable system which can accommodate syncopation and drum fills which might easily cause erroneous judgements. Whilst sensitivity to tempo change and responsiveness are desirable attributes, we also require an appropriate tension between the system and the drummer that resists tempo change, but is still compliant when required. During his test, Rocco Webb had to try to slow its tempo from a false start at 170bpm to 120bpm. He agreed with Joe Caddy that the response could be faster, but also suggested that "*the advantage is that in any musical situation, you'd never want to slow down that much like I did in the course of three minutes. You don't want a hair trigger.*"

3.5 Extending the system

B-Keeper provides a metrical structure for processing information within Max/MSP. By reference to a song structure, changes within the system's own parameters and externals MIDI and data messages can be sent to control other systems. For instance, we have seen how in Ableton Live it is possible to arm tracks and record audio by sending MIDI messages from Max/MSP that are routed via Live's MIDI map to perform the respective functions. Through the use of appropriate MIDI controllers, a complex network of commands can be designed to control different sections of pre-recorded or looped audio, which will lock in time with to drummer.

Jitter [3] is a powerful video manipulation environment within Max/MSP. Messages within Max can be used to synchronise audio and video accompaniment to the music. Lighting can be controlled via DMX messages which can be sent using ethernet-to-DMX or MIDI-to-DMX converters. Using MIDI-to-voltage converters such as Miditron [4], Electrotap's

[3] `http://www.cycling74.com` as viewed 14th April 2009
[4] `http://eroktronix.com/` as viewed 14th April 2009

Teabox [5] or the Arduino [6], it is possible to send voltage to external electronic circuits so that events happen that are synchronous with the drums. In particular, this allows the system synchronise motorised robotics playing musical instruments to drums.

3.6 Summary

There is an acknowledged trade-off between reactiveness and inertia [GD05], applicable to beat tracking algorithms. If a system is reactive, new data is acted upon quickly, so the system is ready to change. Inertia determines how much past data affects the system. If previous estimates are contradicted by new information, the challenge is how to make the correct change and interpretation of new data is a key factor.

The B-Keeper system reacts in two ways. By deciding whether the information is to be acted upon, only data that supports the current hypothesis is used and unreliable observations, of which there can be many, are discounted. The second reaction maintains the balance required for this process to be effective. If observations are outside our window of expectation, it suggests that the window could be too narrow and so even if the observation causes no tempo adjustment, the system updates its parameters for the next observation. There is an inevitable *latency* in reaction time. The greater the change in the local or global tempo, the more latency will be experienced before the system's parameters have updated sufficiently to use this information. Change can be accommodated up to a certain point, beyond which, the design of the system cannot correctly interpret the data.

We have developed a model of drum tracking that incorporates the two-process tempo and phase model suggested by psychological experiments into human tappers [Rep05]. The rapid phase correction observed in humans is mirrored by a dedicated phase-correction process in our drum tracker, which synchronises to salient beats of the bar. By re-adjusting the phase over a period of two to four beats, any local tempo changes are

[5] `http://shop.electrotap.com/products/teabox` as viewed 14th April 2009
[6] `http://www.arduino.cc/` as viewed 14th April 2009

quickly accommodated by the system, whilst the effect to the underlying period is minimal.

Our system is based on the entrainment model which is the basis for many approaches to the problem of beat tracking. It is particularly applicable to drums due to the observed regularity of pulse that is seen in these signals. From the initial reports, it is clear that drummers experienced a real interaction with the system, albeit not the same as the interaction with a human musician. We will now investigate ways to evaluate our system, looking at objective, quantitative data and devising an experimental set-up in order to evaluate the subjective experience of interaction.

Chapter 4

Evaluation

Evaluation of this type of system is not straight-forward since it is intended for use in live performance. There are two areas of evaluation study which may be applicable to B-Keeper. In the field of music information retrieval, beat tracking algorithms have been evaluated offline by testing on a large database of stereo recordings of music from several genres. In human computer interaction (HCI), task-based studies have been used to evaluate musical interfaces. First, we shall look at the methodology and the kinds of quantitative measurements used for evaluating previous beat tracking algorithms.

4.1 Evaluation of Beat Tracking Systems

Goto and Muraoka [GM97] presented a quantitative method for beat tracking evaluation whereby the beats placed by the algorithm were compared with those placed manually by a human annotator which act as 'ground-truth' data. In order to ensure the annotator placed beats as accurately as possible, they had access to a visual sound file and could re-edit the beat locations. A beat is labelled 'correct' if it is within 17.5% of the beat period from the annotated beat. This boundary is based on the limitations of establishing a ground-truth by tapping. Goto and Muraoka then determine the *longest continuous segment* (LCS) to be tracked, for which the mean and standard deviation of the difference must be within 20% of the tactus interval. A minute-long excerpt is tracked correctly if it begins no more than 45 seconds after the start and continues to the end.

The LCS is defined as the ratio of the longest segment to the complete excerpt.

Goto's method of detecting 'correct' beats was adopted for the MIREX 2006 [MMDK07] competitions, where the 'P-score' is calculated by counting of how many beats were within 20% of the inter-tap interval length from the location of the annotated beats (thus the total area which is counted as correct is 40% of the tactus). The P-score is then averaged over the forty sets of annotations from different human tappers. The best algorithm, Dixon's BeatRoot, has a P-score of 0.57 over the dataset, where the humans tappers collectively achieve a score of 0.63. This shows the extent of disagreement within humans on the tapping task. The majority of these discrepancies occur when the chosen tempo is twice or half the most commonly agreed rate.

Klapuri [KEA06] adopted the use of the LCS, an evaluation measure suited to offline beat tracking algorithms since it tests the dual requirements that the error is relatively low and that the algorithm is consistent over time. The excerpts used were approximately one minute in length and he contrasted a causal and non-causal algorithm with implementations of Dixon's and Scheirer's algorithms. Discontinuities in tempo are explicitly excluded by requiring that there are no tempo changes of more than 40% per beat. The results varied across genres, with the LCS being over 90% for rap and hip-hop, whilst it measured between 40 and 60% for jazz and classical. The algorithms tend to perform moderately well on rock and pop songs with between 75% and 85% LCS. Given the initial time taken at the beginning of the excerpt to 'stabilise', these results suggest that Klapuri's comb filter method could be applicable to online systems if the computation time is reduced, perhaps by considering a smaller range of tempo hypotheses. It is instructive to listen to the audio examples on Klapuri's website [1], both where the algorithm is successful, and to hear examples such as Glenn Miller's 'In The Mood', where the phase inverts in an unexpected manner.

Music recorded within the last two to three decades, in particular hip

[1] `http://www.cs.tut.fi/~klap/iiro/meter/examples.html` as viewed 9th May 2009.

hop, rap, rock and pop, may have been recorded with a fixed tempo. The inclusion of a drum machine in a song, or use of a 'click track', has been increasingly popular as the digital audio workstation (DAW) has become an alternative recording medium to analogue tape in the studio. Although this ought to significantly affect a beat tracker's performance, there has been little or no investigation of the effect this has on beat tracking evaluation.

4.2 Measuring Timing Accuracy in B-Keeper

Considering the measures used to evaluate other beat tracking systems, it is apparent that they require adaptation in order to test B-Keeper. Use of the longest continuous segment would be appear to be a promising measure, but in practice, musicians are interacting with the accompaniment and continually adapting to the behaviour of the machine. To run such a test efficiently, we would have to use offline data or a scenario in which the drummer no longer hears the output of the system. This fails to evaluate an important aspect of the beat tracker, namely its behaviour in an interactive context. The style of the piece determines the difficulty of the tracking procedure so that for our system, when tracking rock and dance drums, we would expect the LCS to be close to 100% using the metric of Goto and Muraoka. Also the benchmark of 17.5% of the tatum period from the beat, corresponding to almost 100ms at 120 , is much wider than our goal for perceptual synchrony of 30ms. Thus, according to offline evaluation measures, a performance system could be evaluated as correctly beat tracking a live drummer, yet not be suitable for generating an accompaniment unless it was purely harmonic, with no distinctive rhythmic contribution. Although we cannot carry out a satisfactory comparative evaluation, it is instructive to look at quantitative data as a measure of the success of the alignment algorithm and we shall undertake an alternative investigation of the interactive nature of the system later in this chapter.

It is possible to form some objective measure of the system's performance by examining its response to input signals from live drums. We can

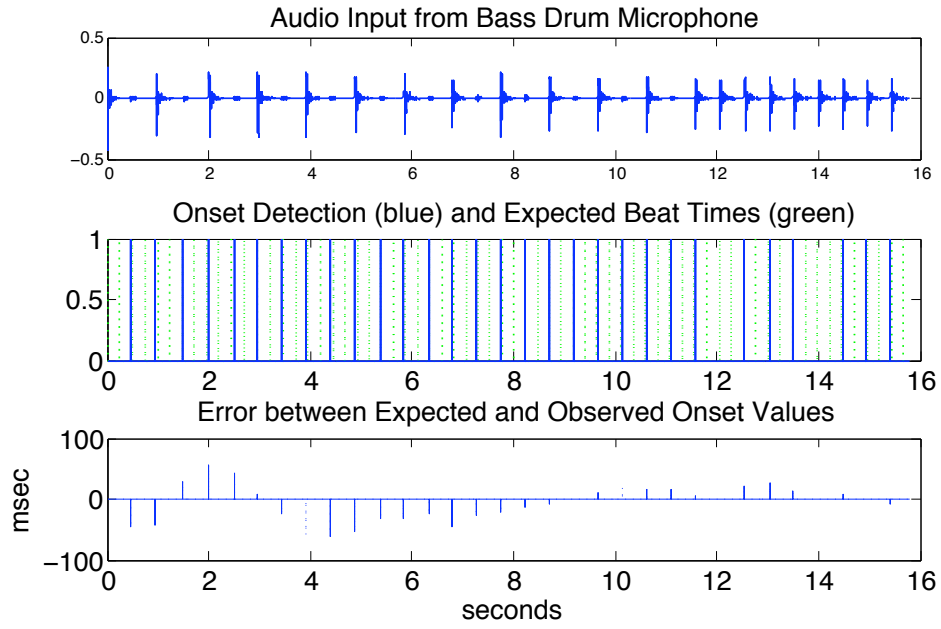

Figure 4.1: Diagram showing B-Keeper's initial response to a bass drum signal with simple pattern.

identify how the BPM is adjusted in order for the sequencer to remain in time and measure the resulting error between the click track which represents the system's expected beat times and the onset times that occur. There is an inevitable error due to player inaccuracies, so that any beat tracking system must attempt to find a line of best fit through onset points which are only approximately in time. Thus, for any given input, there is no definitive response, however, a measure of success can still be given by examining the error and other performance data.

In Figure 4.1, we can see how an early version of B-Keeper responded to a regular 4/4 Kick drum pattern, typical of dance music. David Nock, a session musician who has played on records by rock band 'The Cult' and with dance group 'The Orb', played the drums, and the sole input to the system was from the bass drum. The pattern remained a simple 4/4 dance beat through-out. The tempo estimate is slightly inaccurate at the beginning, causing higher error measurements whilst adjustments are made. Within four bars, the system made adjustments which bring the error to within around 20ms. In Figure 4.2, we can see the error times

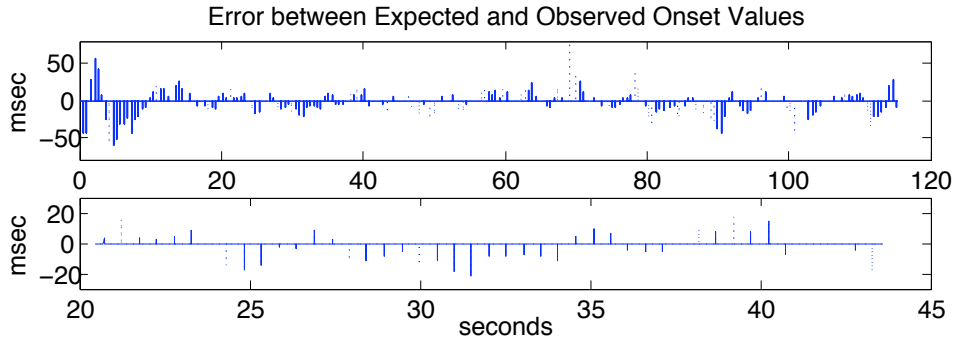

Figure 4.2: The error times over a full performance by David Nock (top) and magnified over a short extract (bottom). Solid error measurements correspond to onsets used by the algorithm in synchronisation whilst dotted error measurements were not used.

from the synchronisation algorithm for the full performance. The mean error of onsets used to synchronise the system is 10.2ms. Given that events occurring within 20 to 30ms are perceived by the human auditory system as being synchronous [LK04], this is the kind of error we require for use within performance.

In Figure 4.3, we can see how the system makes large synchronisation adjustments to rectify discrepancies between prediction and observation during a piece played by Mark Heaney, current drummer with 'The Gang of Four'. Synchronisation adjustment takes place for the onsets corresponding to solid error measurements whereas the dotted errors were ignored. The measurements resulting in phase re-estimation are preceded by a section of increasing error measurements (dotted) which change the parameters of the algorithm but are initially ignored as error. At the point of re-synchronisation, the threshold has decreased sufficiently for the system make an adjustment. After a couple of bars, the error has been rectified in conjunction with the tempo tracking algorithm and there is synchronisation within approximately 10ms again.

4.2.1 Timing Measurements from Performance

Having analysed data from a small number of performances, we recorded the following measures of the error between onsets used to synchronise the

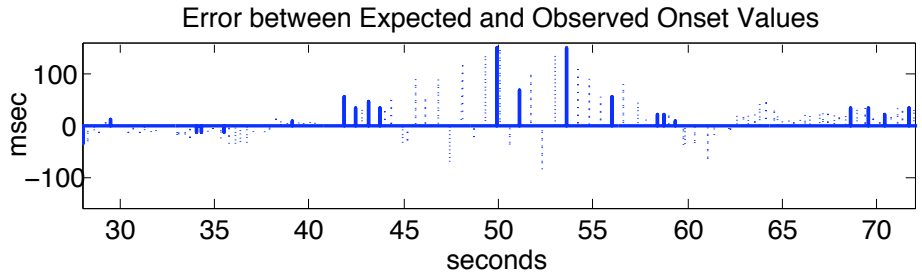

Figure 4.3: B-Keeper's response to a difficult passage with drummer Mark Heaney.

Drummer	Beats Counted	Mean Error	R.M.S.Error
D.Nock	176	10.2	15.2
R.Webb	120	16.0	27.0
J.Caddy	165	16.7	22.5
A.Pickard	202	19.0	26.9
M.Heaney	74	27.0	38.1

Table 4.1: Table showing the mean and root mean squared error encountered in performances by five different drummers.

beat tracker and their expected times for performances by five different drummers.

The observed errors are generally within our desired window of 30 msec for which events will be perceptually synchronous. However, the interpretation of these measurements is not as straightforward as for offline beat tracking, where the error is between the algorithm's predicted beat locations and 'ground-truth' human annotations. Here, the measurement is relative to actual beat events, and hence an element of error is unavoidable in an expressive performance and is dependent upon playing style. A 'zero-error' performance would correspond to a drum machine at a fixed tempo. The larger mean error recorded from Mark Heaney is probably a result of the increased amount of *push* and *pull* against a 'straight' rhythm that was present in his playing. In adjusting the machine to work with his playing style, we found it was necessary to employ a manually set window size in the synchronisation algorithm and set this to be fairly low. That way, his expressive fills did not throw the system out of time.

4.2.2 Discussion

We can make use of quantitative testing to illustrate the behaviour of our drum tracking system. However, there is no comparative test, analogous to the MIREX test for offline trackers, available to us. Partly, the problem is that we have made specific assumptions on the nature of the signal, so that existing datasets, such as those used in MIREX, are not within the scope of the system. The other problem is that we wish to test the system in the context for which it has been designed: as a performance tool for automatic accompaniment. Although it is possible to rate the tracker both in live performance and with offline drum recordings, these tests tend to tell us more about the behaviour of the tracker than provide a useful quantitative measurement. Through iterative development, the drum tracking algorithm improved to the extent that it can regularly be expected to score 100% on the longest continuous segment; but this score is only informative if it is contrasted with other tracking systems.

In evaluating a real-time performance system, we wish to test not just quantifiable data such as beat prediction times, but the subjective experience of interaction. Given the two-way nature of the interaction process, the system influences the playing of the drummer and vice-versa, so that measurements of statistical data, whilst useful for analysis, are also dependent upon the behaviour of the drummer and need to be understood in that context. If a drummer chose to play *to* a metronome, the the metronome would make extremely good predictions, but it would not succeed at being an interactive system since it fully determines the tempo. Thus, although the difference between the predicted and observed time of events provides a direct quantitative measurement of the system's performance, we must be careful when interpreting this data, since these measurements are dependent upon the playing style.

The accuracy measurements indicate that our beat tracker achieves the bound of 20 to 30ms recommended for interactive systems [LK04] as discussed in Chapter 2. It also scores 100% LCS according to the metric proposed by Goto and Muraoka. However, the situation is very different to that confronting offline beat trackers since the drummer is able to listen

and respond to auditory feedback from the system. We now look for an evaluation strategy that can go beyond quantitative measurements in order to give a reliable indication of how well B-Keeper functions as a live performance tool.

4.3 Evaluation of Musical Interfaces

In order to develop a design for evaluation, we will look at evaluation methodologies for musical interfaces and discuss the nature of the interaction we are trying to measure. Human Computer Interaction (HCI) is concerned with the design, evaluation and implementation of interactive computing systems for human use. Evaluation methods have traditionally been task-based, using task analysis, cognitive walkthrough and usability studies to evaluate the completion or efficiency of specific tasks. Fitt's law, which measures the trade-off between speed and accuracy when pointing at a target, has been shown to apply to traditional HCI tasks, such as target acquisition. It has been used by HCI to compare different input devices, transforming performance scores into indexes of performance corresponding to the devices and independent of the actual experimental set-up. Many of the tasks used in HCI have been formulated for the design of graphical user interfaces.

Wanderley and Orio [WO02] adapted this task-based approach for the evaluation of new musical interfaces. Musical controllers are a specialised category of HCI input devices, in which several parameters are controlled separately using sensory feedback from the body. Time has a privileged relationship for musical controllers, since the user requires high precision *in* time, whereas for most HCI tasks, time is a variable that is measured for a task. They proposed that controllers could be evaluated by requiring that *musical tasks* be completed, such as the control of pitch, musical gestures, such as glissando or trill, and the replication of musical phrases. Wanderley and Orio contrast the subjective nature of musical interfaces to the more measurable objectives of traditional HCI, which might place more emphasis on task efficiency:

> "the question here is whether this measurement must necessarily be quantitative, as in the case of HCI. In music, it must be noted that controllers cannot be evaluated without taking into account subjective impressions of performers, ruled by personal and aesthetic considerations.'

The drum tracker could be considered a musical controller to the extent that drums are used 'to control tempo'. However, it has not been designed *as* a tempo controller suited to performing specific tempo-related tasks. Rather, we expect that the tempo will be controlled by the drums through a similar interactive process as that which occurs between musicians. We now describe an evaluation methodology which can be used for B-Keeper in an interactive context: the Turing Test.

4.4 Evaluation of B-Keeper using a Turing Test

Offline beat trackers are evaluated relative to annotations of human tappers who provide a ground truth against which they can be judged. This is necessitated by the fact that the beat is a perceptual construct, so we require humans to indicate the 'correct' beat locations. Since the beat is only defined to a limited precision, the annotations for these tests are determined either collectively or by using an 'expert' tapper who annotates offline using audio editing software.

For a real-time beat tracker, it would therefore make sense to aim to compare the algorithm with a human tapper, but we face the dilemma of how best to do so. Since the drum tracker controls an accompaniment to synchronise with a musician, ideally, we wish to evaluate the result of this interaction and contrast it with the interaction occurs if a human controlled the accompaniment. In order to do so, we have looked to the paradigm of the 'Turing Test' in which an interrogator is asked to distinguish between human and machine.

4.4.1 The Turing Test and Variants

In Alan Turing's 1950 paper, 'Computing Machinery and Intelligence' [Tur50] he proposes replacing the question "can a computer think?", by an *Imitation Game*, popularly known as the "Turing Test", in which it is required to imitate a human being. Turing formulates the problem as a game in which a man pretends to be a woman, and an interrogator is asked to distinguish between them on the basis of typewritten answers alone.

> "We now ask the question, 'What will happen when a machine takes the part of A in this game?' Will the interrogator decide wrongly as often when the game is played like this as he does when the game is played between a man and a woman? These questions replace our original, 'Can machines think?' "

The attribution of mind or mental states to the machine is thereby based on a functional test: our ability to discriminate between man and machine on the basis of empirical evidence alone. Turing considered many objections to this philosophical position within the original paper and there has been considerable debate as to its legitimacy, particularly the position referred to as 'Strong A.I.'. Famously, John Searle [Sea80] put forward the Chinese room argument which proposes a situation in which a computer might be able to pass the test without ever *understanding* what it is doing. In this example, it can hardly be said that the computer "has intelligence", it merely carries out mechanical instructions, such as looking words up in a dictionary, in order to reply. Harnard [Har00] discusses the implications of Turing's paper for empirical research on minds and machines, describing a hierarchy of Turing tests, and acknowledges the validity of Searle's argument in the limited domain of machines which function exclusively in a symbolic domain. After his defeat by chess-playing program Deep Blue in 1997, Gary Kasparov expressed scepticism about whether the computer had indeed been acting alone, leading Krol [Kro99] to claim that the computer had become the first to 'pass the Turing Test'. This is an example of a symbolic domain for game playing in which a computer has been able to rival and surpass the intelligence of

an expert human.

Cohen [Coh05] criticises the test as being focused on logical, interpersonal, linguistic intelligence as opposed to internal, visual-spatial or musical intelligence. However, the scenario of the Imitation Game might prove to be an interesting model for constructing an experiment to evaluate an interactive musical system. Whilst we do not wish to claim the system possesses 'intelligence', its ability to behave *as if* it had some form of 'musical intelligence' is vital to its ability to function as an interactive beat tracker.

The test has been applied to several challenges in the development of human-computer interaction. The Loebner Prize [Shi94] is an annual competition which is based on Turing's original formulation, requiring contestant algorithms to converse with a judge via an interface. When the test requires a response to words with semantic meanings, the common result is that the programs do not perform well. Laird and Duchi [Lai00] first proposed the Turing test for the game Quake II, in which human players were correctly identified 89% of the time and computer bots correctly identified 56% of the time. The Computer Game Bot Test [2] was subsequently held as part of the IEEE Symposium on Computational Intelligence and Games 2008, and asked judges to discern between a human player and a computer-controlled 'bot' for the game 'Quake-3'. In neither the Loebner Prize nor the computer contests has an algorithm yet succeeded in convincing the judges. Livingstone [Liv06] identifies the challenge of creating believable AI in computer games as successful role-playing by computer characters, both individually and collectively, to mimic the tactics and style of human ones. He regards Harnard's hierarchy of Turing tests as applicable to computer games, where the passing of a t1 test, a toy-version of the test which applies to sub-routines, would result in realistic game experiences.

The problem of evaluating the contribution of a computer to in an aesthetic medium has been addressed by Pearce and Wiggins [PW01], who

[2] `http://botprize.org/` as viewed 15th April 2009.

look at evaluation methodologies used in a range of algorithmic composition programs. Often the evaluation offered is merely a stated subjective opinion on the part of the system's developers. Seeking a scientific basis for their claims, they recognise the need for an experiment that has the potential to 'falsify the theory' in the manner famously proposed by Karl Popper [Pop59]. They propose a framework which has objective validity and removes the developers' judgement from the process. The evaluation stage takes a similar form to Turing's "imitation game", where a combination of human and computer-generated pieces of the proposed style are played to participants in the experiment. Pearce and Wiggins reduce the claim made by the test from that of machine "intelligence" to assessing membership or non-membership to the set of human-created pieces. Whereas in Turing' s original conception the interrogator is free to interact with the computer, when used for evaluating algorithmic composition the test becomes more a *discrimination game*. Pearce and Wiggins provide details of an experiment in which a drum and bass composition program is formally evaluated according to the proposed framework. Particular care is taken to remove the developer from the evaluation process and to analyse the results for statistical significance. The algorithm's failure to pass the test serves to illustrate it's usefulness as a benchmark for successful algorithmic computational creativity.

David Cope [Cop01] describes a comparison test known as "The Game", in which 'computer-generated' pieces in the style of Chopin by his *Emmy* algorithm are contrasted with lesser-known, but musically 'average' (as opposed to exemplary or weak) pieces by the composer. The object of "The Game" is to pick out the computer-generated pieces. Cope interacts with the algorithm in order to create the computer's piece, thus the algorithm is not fully autonomous. Douglas Hofstadter expressed surprise and alarm at the outcome of this 'Game', which caused him to question the core of his beliefs about the relation between musical expression and the human mind. A key difference between a musical Turing Test and a linguistic one is pointed out by Justus [Jus02] in his review, 'A Musical Jabberwocky', of David Cope's book 'Virtual Music'. The parallel is made with Lewis Carroll's famous poem which features many nonsensical words,

but whose syntax and structure are classically correct. Where language requires sentences to be both syntactically and semantically correct in order to be meaningful, by generating musical phrases that incorporate the correct *syntax*, it appears that elements of *semantic* meaning are thereby present too.

On his website [3], François Pachet describes a musical Turing Test in which jazz pianist, Albert van Veenendaal, played piano with the Continuator system. Two critics, Henkjan Honing and Koen Schouten, had to decide which of the two were responsible for the musical output. The result is presented informally: "The results were largely in favour of the Continuator". The Rencon competition [HBHK04] seeks to compare musical expression implemented by a performance rendering system. Previous audiences considered the renderings "were good", and the organisers proposed to introduce comparative testing, where pieces were ranked for human-ness (the 'Turing Test') and for machine-ness (the 'Gnurit Test').

Where the computer can conceivably take the place of a skilled human, the formulation of the test can quantify the aesthetic impressions of listeners in an unbiased way. Despite the difficulties involved in passing the original formulation of the test, when the same principle is applied in a different discipline, the test has successfully been used to compare human and computer performance and highlight differences between them. In order to evaluate B-Keeper, we designed a large-scale musical Turing Test. Since the participant interacts with the system, the experiment is closer to being a genuine Turing Test than the evaluation method described by Pearce and Wiggins for composition algorithms, since the drummer is free to *interact* with the tempo controller during the experiment. Whilst acknowledging the subjective nature of musical engagement, this framework means that it is still possible to make objective statistical claims about the interaction with the beat tracker.

By setting up a human tapper as one accompaniment controller and B-Keeper as the other, we can directly compare them. When designing the experiment, we also felt that it was important to have a metronome

[3] `http://www.csl.sony.fr/~pachet/Continuator/` as viewed 8th May 2009

controller, maintaining a steady tempo throughout, with which there is no interaction. This functions as a 'control' for the experiment.

There are some important features of the Turing Test that make it so suitable for the evaluation process. We can ask subjective questions of musicians and observers, such as "how well would you rate that?", in a formalised context which can be used to directly contrast the system with a human controller. In order to ensure impartiality, we constructed the experiment to be *double-blind*, so that neither the researchers nor the participants knew which controller was being used until the end of each set of trials. Ordinarily, such questions can remain inconclusive, with the researchers seeking a validation of the system in the absence of any comparative standard. However, in a context where the identity of the controller is hidden and contrasted against a human 'player', a more truthful assessment can be obtained. The Turing Test is thus used to decide two questions: Can the participants distinguish the *identity* of the computer from the others? How well do the participants *compare* the computer relative to the others?

4.4.2 Experimental Design

The test involves a drummer playing along to the same accompaniment track three times. For each test, the drummer gives four steady beats of the kick drum to calculate an initial tempo and start the accompaniment. Each time, a human tapper taps the tempo on the keyboard to keep time with the drummer, but only when the human tapper is selected as the controller will this alter the tempo of the accompaniment. A Gaussian window is applied to the intervals between taps in order to smooth the tempo fluctuation, set so the resulting accompaniment is musical in character, but responds very quickly to tempo change. During the fixed tempo trials, the accompaniment remains at the initial tempo and during the other trial the tempo is controlled by B-Keeper. An illustration of the experimental design is shown in figure 4.4.

We are interested in the interaction between the drummer and the accompaniment which takes place through the machine. In particular, we

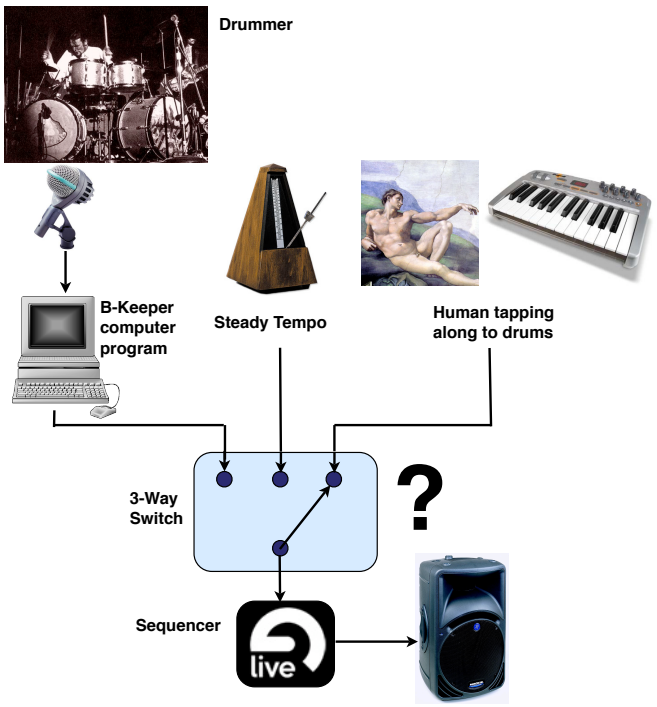

Figure 4.4: Design set-up for the experiment. Three possibilities: (a) Computer controls tempo from drum input; (b) Steady Tempo; (c) Human controls tempo by tapping beat on keyboard.

wish to know how this differs from the interaction that might take place with a person, or in this case, a human beat tracker. We might expect that, if our beat tracker is functioning well, the B-Keeper trials would be 'better' or 'reasonably like' those controlled by the human tapper. We would also expect them to be 'not like a metronome' and hence, distinguishable from the Steady Tempo trials. These expectations will form the basis of our hypotheses that are to be tested and we collected quantitative and qualitative data in order to do so.

After each trial, we asked each drummer to mark an 'X' on an equilateral triangle which would indicate the strength of their belief as to which of the three systems was responsible. The three corners corresponded to the three choices and the nearer to a particular corner they placed the 'X', the stronger their belief that that was the tempo-controller for that particular trial. Hence, if an 'X' was placed on a corner, it would indicate certainty that that was the scenario responsible. An 'X' on an edge would

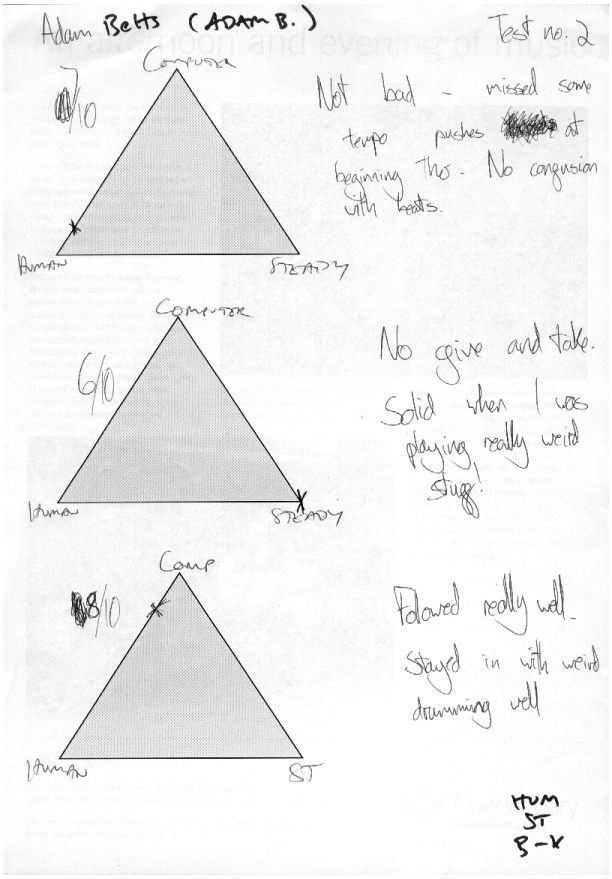

Figure 4.5: Sample sheet filled in by drummer Adam Betts.

indicate confusion between the two nearest corners, whilst an 'X' in the middle indicates confusion between all three. This allowed us to quantify an opinion measure for identification over all the trials. The human tapper (AR) and an independent observer also marked their interpretation of the trial in the same manner. In addition, each participant marked the trial on a scale of one to ten as an indication of how well they believed that test worked as 'an interactive system'. They were also asked to make comments and give reasons for their choice. A sample sheet from one of the drummers is shown in Figure 4.5.

We carried out the experiment with eleven professional and semi-professional drummers. All tests took place at the Listening Room of the Centre for Digital Music, which is an acoustically isolated studio

space. Each drummer took the test (consisting of the three randomly-selected trials) twice, playing to two different accompaniments. The first was based on a dance-rock piece first performed at Live Algorithms for Music Conference, 2006. The second piece was a simple chord progression on a software version of a Fender Rhodes keyboard with some additional percussive sounds. The choice of accompaniment will have an effect on the drummer's playing style and we chose these two pieces so they were suitable for a straight-forward drum pattern and for variation using syncopation and fills. Whilst the human tapper was visible to the drummer, we do not consider the visual communication channel to have had a large effect on the ability of the drummer to distinguish between controllers. It may help the human tapper to be able to use both auditory and visual information follow the drums, but such information is often available to musicians in a live performance. We recorded all performances on video and audio and stored data from the B-Keeper algorithm. This allowed us to see how the algorithm processed the data and enabled us to look in detail at how the algorithm behaved and monitor how the tempo of the accompaniment was changed by the system.

4.4.3 Results

We shall contrast the results between all three tests, particularly with regard to establishing the difference between the B-Keeper trials and the Human Tapper trials and comparing this to the difference between the Steady Tempo and Human Tapper trials. In Figure 4.6, we can see the opinion measures for all drummers placed together on a single triangle. The corners represent the three possible scenarios: B-Keeper, Human Tapper and Steady Tempo with their respective symbols. Each 'X' has been replaced with a symbol corresponding to the actual scenario in that trial. In the diagram we can clearly observe two things:

There is more visual separation between the Steady Tempo trials than the other two. With the exception of a relatively small number of outliers, many of the steady tempo trials were correctly placed near the appropriate corner. Hence, if the trial is actually steady then it will probably be

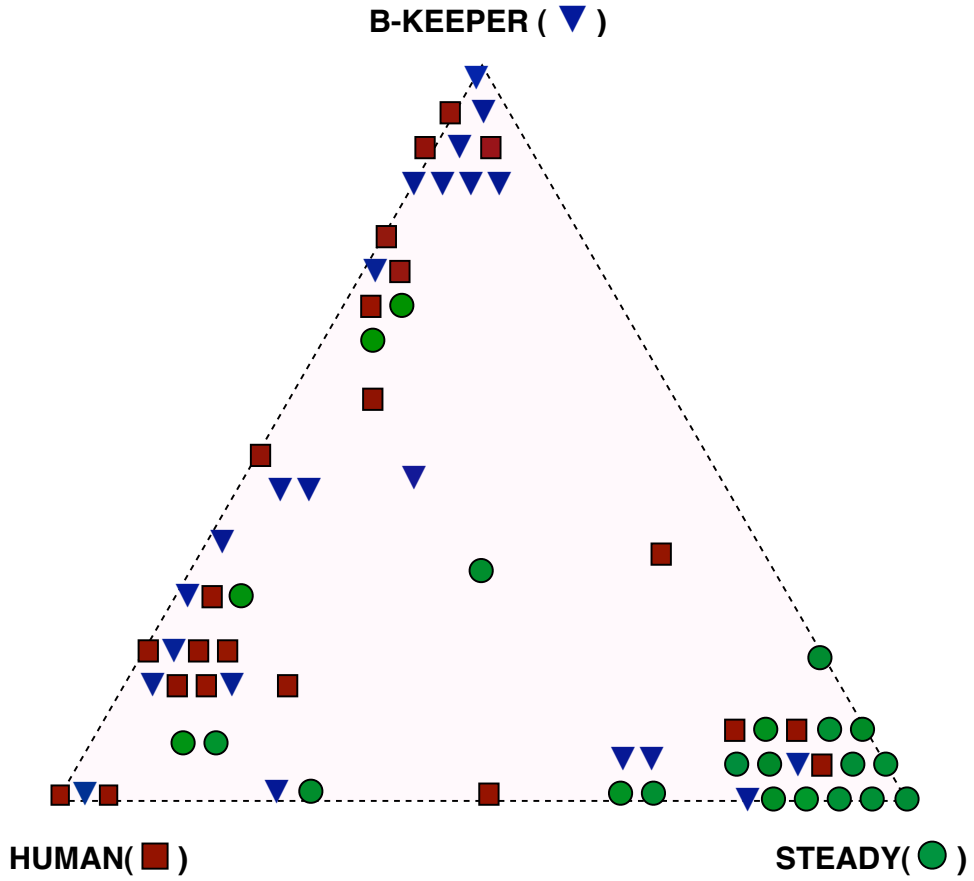

Figure 4.6: Results where the eleven different drummers judged the three different accompaniments (B-Keeper, Human Tapper and Steady Tempo) in the test. The symbol used indicates which accompaniment it actually was (see corners).

identified as such.

The B-Keeper and Human Tapper trials tend to be spread over an area centered around the edge between their respective corners. At best, approximately half of these trials have been correctly identified. The distribution does not seem to have the kind of separation seen for the Steady Tempo trials, suggesting that the drummers had difficulty telling the two controllers apart, but could tell that the tempo had varied.

The deduction process used by participants generally involved them first trying to determine whether the tempo had been steady or not. In the majority of cases, this was successful, but some misidentifications were made, particularly if the drummer had *played to* the accompaniment and

not made much attempt to influence the tempo. In these cases, the distinction between an interactive accompaniment, which will adapt to you, and one at a fixed tempo is harder to judge.

The second deduction to be made would be, in the case where the tempo varied or the music appeared responsive, to discern whether the controller had been B-Keeper or the Human Tapper. In order to do so, there needs to be some assumption as to the characteristics that might be expected of each. From interviews, we recognised that drummers expect the human to be more adaptable to changes in rhythm such as syncopation and they may also have felt that a human would respond better to changes within their playing. For instance, as drummer Tom Oldfield commented: "I felt that was the human, because it responded very quickly to me changing tempo."

Case Studies

Joe Caddy

One dialogue exchange with Joe Caddy, an experienced session drummer whose credits include hip-hop band Captive State, shows the kind of logical debate in action.[4]

JC: [talking about the trials]: "The first one I gave 8 and I put actually closer to human response. I played pretty simply and it followed it quite nicely. The second one had no response at all to tempo on the drums. The last one I gave 9 - great response to tempo change, I slowed it up, I slowed it down. It took a couple of beats to resolve, but I think I put it nearer the B-Keeper."

AR: "Is that because you have some experience of the system?"

JC: "If it was human, I would have expected it to catch up more quickly. I think because it took two or three beats to come in at the new tempo, it was the B-Keeper."

AR: "Same. I think it's an 80 per cent chance that that was B-Keeper."

[4] **JC** refers to drummer Joe Caddy; **AR** refers to the first author, who acted as the Human Tapper in all experiments.

Figure 4.7: AR taps on the keyboard in time with drummer, Joe Caddy, during one of the tests.

[Result is revealed: The first was B-Keeper; the last the Human Tapper, i.e. controlled by **AR** - the opposite to what both **JC** and **AR** have identified.]

AR: "I just didn't think it was that though. I guess it must have been."

JC: "The last test we did, I changed the tempo much more. Do they surprise you those results?"

AR: "The first I felt was me and I felt that the last wasn't me."

This exchange demonstrates how both a drummer and even the person controlling the tempo can both be fooled by the test. From the point of view of the key tapper, AR suggests that there is a *musical illusion* in which, by tapping along to the drummer playing, it can appear to be having an effect when in fact there is none. This illusion was stronger when the B-Keeper system was in operation as the music would respond to changes in tempo. This effect is reflected in the opinion measures reported by AR, which we initially expected to be considerably higher for the Human Tapper trials than the others, but had a mean of only 45% (see Table 4.2).

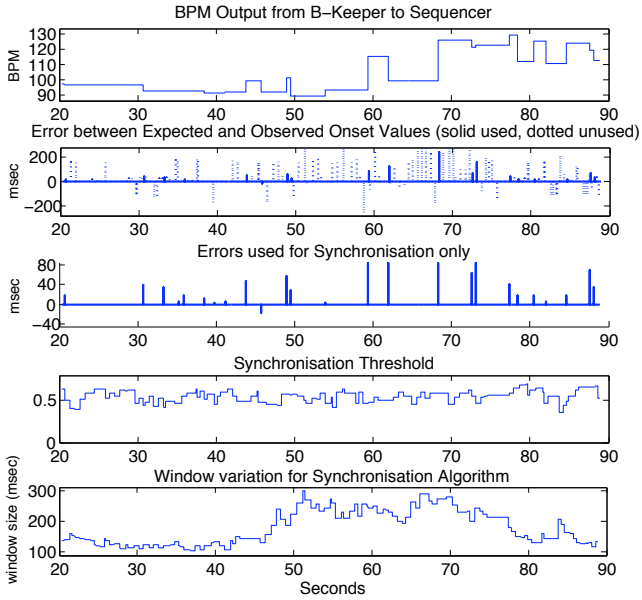

Figure 4.8: Data from the B-Keeper's interaction with drummer Adam Betts. The top graph shows the tempo variation. The second graph shows the errors recorded by B-Keeper between the expected and observed beats. The final two graphs show how the synchronisation threshold and window automatically adapt, becoming more generous when onsets fail to occur in expected locations.

Adam Betts

The above study shows a scenario in which the B-Keeper was mistakenly identified by the drummer (and the human tapper) as being the human-controller. In one trial with James Taylor Quartet drummer, Adam Betts, the machine had been calibrated to the standard setting, so as to be fairly responsive to tempo changes. However, when he played a succession of highly syncopated beats, the algorithm responded by making the synchronisation window so wide that the machine was thrown out of sync. In Figure 4.8, this can be seen happening after about fifty seconds, where the pattern has changed so the onsets are no longer used by the tracker to synchronise (dotted errors in second graph). When it eventually does so at sixty to seventy seconds, an erroneous adjustment easily occurs due to the size of the window and lower threshold.

In this case, it was immediately apparent that the controller was B-Keeper since the tempo had varied and done so in a non-human manner.

It had made an apparent mistake and all three involved in the experiment, the drummer, the human tapper and our independent observer, immediately concluded that this was B-Keeper. On the trial sheet, Adam commented:

> "Scary. Okay at beginning, but got confused and guessed tempo incorrectly with sixteenths etc. When it worked, it felt good."

Such an event happened only one time out of the the twenty-two tests [5], but it is interesting since it suggests that the form of the experiment is viable for similar reasons to those suggested by Turing. In the scenario of the imitation game, if the machine did exhibit abnormal behaviour (for instance, as he suggests, the ability to perform very quick arithmetical calculations) or, as implied throughout Turing's paper, the inability to answer straight-forward questions such as the length of one's hair, then one could easily deduce it was the machine. In this case, the absence of human tolerance to extreme syncopation is the the kind of 'machine-like' characteristic that made it easily identifiable. In the language game, where the Turing Test is usually applied, the 'machine-like' quality is often evident. Trained computer scientists can spot the times when answers are 'triggered' by key words.

Analysis and Interpretation

The mean scores recorded by the drummers are given at the top of Table 4.2. They show similar measures for correct identification of the B-Keeper and Human Tapper trials. Both have mean scores of 44%, with the confusion being predominantly between which of the two variable tempo controllers is operating. The Steady Tempo trials are identified the majority of the time and have a mean confidence score of 64% on the triangle.

Each participant in the experiment had a higher score for identifying the Steady Tempo trials than the other two. It appears that the Human Tapper trials are the least identifiable of the three and the confusion tends

[5] This was due to incorrect parameter settings for the drumming style in question.

		Judged as:		
Judge	**Accompaniment**	B-Keeper	Human	Steady
Drummer	B-Keeper	**44** %	37 %	18 %
	Human	38 %	**44** %	17 %
	Steady	12 %	23 %	**64** %
Human Tapper	B-Keeper	**59** %	31 %	13 %
	Human	36 %	**45** %	23 %
	Steady	15 %	17 %	**68** %
Observer	B-Keeper	**55** %	39 %	6 %
	Human	33 %	**42** %	24 %
	Steady	17 %	11 %	**73** %

Table 4.2: Mean Identification measure results for all judges involved in the experiment. Bold percentages correspond to the correct identification

to be between the B-Keeper and the Human Tapper. In the trials involving B-Keeper, drummers were least confident about identifying it as the controller. The researchers, who acted as independent observer and the tapper, were more confident. In an analogous result, we might expect the human tapper, the first author, to be able to distinguish the trials in which he controlled the tempo, however, this often did not appear to be the case. He was more successful at discerning the two trials where his actions did not control the tempo.

We can polarise the decisions made by drummers by taking their highest score to be their decision for that trial. In the case of a tie, we split the decision equally. The advantage of this method is that we can make pair-wise comparisons between any of the controllers, whilst also allowing the participants the flexibility to remain undecided between two possibilities. Table 4.3 shows the polarised decisions made by drummers over the trials. There is confusion between the B-Keeper and Human Tapper trials, whereas the Steady Tempo trials were identified over 70% of the time. The B-Keeper and Human Tapper trials were identified in 43% and 45% of cases respectively, little better than chance.

	Judged as:		
Controller	B-Keeper	Human	Steady
B-Keeper	**9.5**	8.5	4
Human Tapper	8	**10**	4
Steady Tempo	2	4	**16**

Table 4.3: Table showing the polarised decisions made by the drummer for the different trials.

	Judged as:	
Controller	Human Tapper	Steady Tempo
Human Tapper	12	4
Steady Tempo	5	14

Table 4.4: Table showing the polarised decisions made by the drummer over the Steady Tempo and Human Tapper trials.

Pair-wise Comparative Tests

In order to test the distinguishability of one controller from the other, we can use a Chi-Square Test, calculated over all trials with either of the two controllers. If there is a difference in scores so that one controller is preferred to the other (above a suitable low threshold), then that controller is considered to be chosen for that trial. Where no clear preference was clear, such as in the case of a tie or neither controller having a high score, we discard the trial for the purposes of the test.

Thus, for any two controllers, we can construct a table that shows pair-wise polarised decisions between them. The table for comparisons between the Steady Tempo and the Human Tapper trials is shown in Table 4.4. We test the hypothesis that the distribution is the same for either controller, corresponding to the premise that the controllers are indistinguishable.

The Chi-Square Test statistic for this table is 8.24 which means that we reject the test hypothesis at the 5% significance level. This indicates a significant separation between the controllers. Partly, this can be explained from the fact that drummers could vary the tempo with the Human Tapper controller but the Steady Tempo trials had the characteristic of being metronomic.

Comparing the B-Keeper trials and the Human Tapper trials, we get the results shown in table 4.5. The Chi-Square test statistic is 0.03 which

	Judged as:	
Controller	Human Tapper	B-Keeper
Human Tapper	9	8
B-Keeper	8	8

Table 4.5: Table contrasting decisions made by the drummer over the B-Keeper and Human Tapper trials.

is extremely low, suggesting no significant difference in the drummers' identification of the controller for either trial. Whilst B-Keeper shares the characteristic of having variable tempo and thus is not identifiable simply by trying to detect a tempo change, we would expect that if there was a *machine-like* characteristic to the B-Keeper's response, such as an unnatural response or unreliability in following tempo fluctuation, syncopation and drum fills, then the drummer would be able to identify it as the machine. It appeared that, generally, there was no such characteristic and drummers had difficulty deciding between the two controllers. It may appear that having the Human Tapper visible to them would give them an advantage, however, this did not prove to be the case as the similarity between the computer's response and a human tapping along was close enough that often the observer and the human tapper were also unsure of the controller.

The difficulty of distinguishing between controllers was a common feature of many tests and whilst the test had been designed expecting that this might be the case, the results were often surprising when revealed. We did not expect drummers to believe that steady accompaniments had sped up or slowed down with them nor the human tapper to believe he had controlled the tempo when he had not. This indicates a subjectivity to the perception of time. It seems that some drummers had an enhanced ability to spot a fixed tempo without even varying much, perhaps gained through extensive experience. Matt Ingram, session drummer, who professed to have been "playing to click for the last ten days, all day every day", remarked of the Steady Tempo trial: "It felt like playing to a metronome, 'cause it was just there. Either that or your time's great, 'cause I was trying to push it and it wasn't letting me."

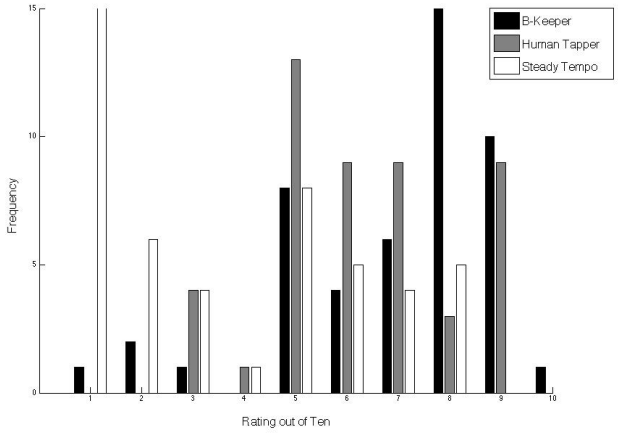

Figure 4.9: Bar Graph indicating the different frequency of cumulative ratings for the three scenarios - B-Keeper (black), Human Tapper (grey) and Steady Tempo (white).

Ratings

In addition to the identification of the controller for each trial, we also asked each participant to rate each trial with respect to how well it worked as an interactive accompaniment to the drums. For an interactive accompaniment to work, it should achieve a close synchrony with the drums and also be response to change. Our reasoning in obtaining ratings for the accompaniments is that in addition to trying to establish whether the beat tracker is distinguishable human tapper controller, it is also desirable to compare the controllers through a rating system.

The cumulative frequency for these ratings over all participants (drummers, human tapper and independent observer) is shown in Figure 4.9. The Steady Tempo accompaniment was consistently rated worse than the other two. The median values for each accompaniment are shown in Table 4.6. The B-Keeper system has consistently been rated higher than both the Steady Tempo and the Human Tapper accompaniment.

The overall median ratings, calculated over all participants, were:
B-Keeper: 8; Human Tapper: 6; and Steady Tempo: 5.
It is important that not only was the beat tracker not significantly distinguishable from the human tapper, but it performed as well when judged by both the drummer and an independent observer. The fact that the median

	Median Rating		
Judge	B-Keeper	Human Tapper	Steady Tempo
Drummer	7.5	5.5	5
Human	8	6.5	4
Observer	8	7	5
Combined	**8**	**6**	**5**

Table 4.6: Median ratings given by all participants for the different scenarios. The combined total median is given in bold.

rating is towards the top end of the scale suggests that musically the beat tracker is performing its task well. As the experiment was double-blind, there was no bias within the scaling of the different controllers.

If we look at pair-wise rankings, we can assess the significance of this difference between ratings. Firstly, we convert the rating out of ten into a strict ordinal rating (allowing equality where necessary). The Wilcoxon signed-rank test, e.g. [SS00], is a non-parametric statistical test that can apply to test the hypothesis that the controllers' rankings have the same distribution. For more than twenty trials, the distribution for this test statistic is approximately normal.

When contrasting the rankings given by drummers to B-Keeper with the Steady Tempo and Human Tapper trials, the approximate Z ratios [6] are 2.97 and 2.32 respectively. Thus, we would reject the hypothesis that the controllers are equally preferable at the 5% significance level in both cases. The fact that the ratings are significantly higher for B-Keeper is highly important as the primary aim is to create a musically successful beat tracker for live drums.

4.5 Summary

In this experiment, we contrast a computer-based beat tracker with a human tapper and metronome for the purposes of providing interactive accompaniment to drums. B-Keeper has proved to be comparable in performance, and aesthetically preferable, to the Human Tapper and is not

[6] normal with zero mean and unit variance

distinguishable in any statistically significant way. The Steady Tempo accompaniment was perceived as a less successful accompanist and was statistically distinguishable from the two variable tempo controllers.

The musical Turing Test has proved a suitable framework for the evaluation of a beat tracking system. It is able to evaluate the subjective phenomenon of musical interaction in a scientific context. Subjective claims that a system works well are replaced by statements of indistinguishability between the system and human that can be statistically tested. Whilst subjective judgements are used within the test, both to identify the system and to rate the quality of the accompaniment generated, the results of the test have an *objective validity*. Since the musician is free to interact with the system during the test, Turing's original formulation is preserved, although transferred from the domain of language to that of music. Turing replaced the question "is the computer intelligent?" by a behavioural test, and similarly, our question "is the computer interacting well?" is replaced by the question "is the resulting interaction distinguishable from that with a human?". The latter question is falsifiable whereas the former relies on subjective opinion. Whilst subjective evidence can support the claim, no number of occurrences of seemingly "good interaction" could prove this to be true. The test might be applicable to other interactive systems; for instance, a score-follower generating a piano accompaniment could be evaluated within the same formulation. In cases where extensive rehearsal is required with the system, this may be difficult to accomplish, however, any musical scenario in which the computer's role could be adequately performed by a human could conceivably be suited to the musical Turing Test.

Chapter 5

B-Keeper: Further Modifications and Evaluation

Having carried out a successful evaluation of the system involving many drummers, we decided to investigate whether adjustments could be made to improve the algorithm. The musical Turing Test did highlight one significant weakness of B-Keeper. Whilst the system worked well for simple beats, a passage of complex syncopation by Adam Betts resulted in rapid parameter adjustment that caused it to fail the trial, and more importantly, to speed up quickly. Jehan [Jeh05] has emphasised the *bottom-up* nature of most beat tracking approaches and their failure on syncopated and polyrhythmic music. This is attributed to a lack of metrical interpretation of the incoming rhythm or *top-down processing*. The anomaly during the trial occurred due to poor top-down interpretation of a complex drum signal and the work described in this chapter aims to identify and rectify this aspect.

One criticism of top-down models is that by using a set of rules, they encode a cultural bias of musical interpretation. However, bottom-up beat trackers may also encode a necessary bias through assumptions about the nature of the signal, such as the implication that significant peaks in the onset detection function correspond to beats. The top-down model enables metrical theory, such as Lerdahl and Jackendorff's GTTM [LJ83], to be applied by a beat tracker so that a rhythmic signal is interpreted in a more exact manner. The synchronisation weights, corresponding to drum patterns, that are used by B-Keeper are a combination of a culturally-specific prior, dependent upon whether the beat is strong or weak as defined by

the GTTM, and the pattern learnt by recent observation. The hierarchical structure proposed in the GTTM and other traditional theories of meter is modelled explicitly in the new developments we describe, which will be illustrated in the context of two case studies that have taken place.

5.1 Case Study A : Hook and the Twin

Hook and the Twin are a new UK band who make use of an innovative live set-up. Tom Havelock plays bass, guitar, synthesizers live, with no pre-recorded parts, making use of Ableton to loop each new section. The songs tend to have linear structures so that the multiple layered instruments work in different combinations. By using a MIDI footpedal, messages are sent to trigger sets of instructions, such as bring in 'all recorded loops', 'erase my last recording', or 'switch to bass'.

Since B-Keeper has been implemented via the use of Ableton Live, it was capable of taking over control of the tempo, allowing drummer Marcus Efstratiou the freedom of not wearing headphones. Testing was generally successful, however, since the band were not only using the system to play back previously recorded loops, they were also using it when recording those loops and so it could not be guaranteed that the sequencer's time was identical to the drummer's.

We experienced two conflicting needs of stability and responsiveness. The trade-off between responsiveness and inertia has been commented on by Gouyon and Dixon [GD05]. The band want the system to respond to tempo changes "without dragging." Marcus Efstratiou stressed the importance of *confidence* in the system. One major requirement is that he has "got to be able to speed up and know that it'll follow me." When this was achieved through a high expander setting, it also resulted in instability since the standard deviation is thereby prone to increasing rapidly during fills and off-beat syncopation. Ideally, we only want the expander to increase when a beat is missed at a high metrical level, or to interpret the input so that it synchronises with the tempo changes without being thrown out of phase by complex patterns. It is possible that the expander setting could be lower and a high beta setting should have been

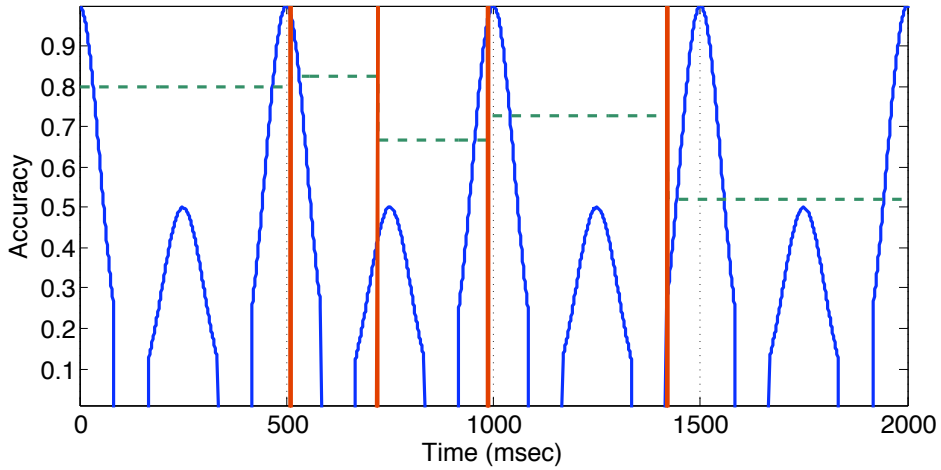

Figure 5.1: Illustration of how the algorithm behaves for events of differing accuracy. The provision for sixteenths can be seen as notches, visible between the Gaussian shaped windows around expected beat locations. The tatum period in this example is fixed at 250ms.

used, which would fully correct any phase discrepancy. Since beta was only 0.4, when phase correction is large but within the window, it only corrects to 40% of the value, providing a more stable accompaniment, but one that is less responsive. For the high expansion setting we used, a discrepancy occurred between the timeclock of the drummer and that of the sequencer. Since the recorded parts are played *to the drummer*, when the synchronisation was varying, the timing error on the resulting loops seemed exaggerated. When these reactive settings are used, the system is highly sensitive to syncopation and becomes more reliable. We made two significant changes to the algorithm incorporating a better top-down interpretation of events.

5.1.1 Sixteenths

Conversations and tests with Marcus Efstratiou and Adam Betts pointed out a weakness with the system. Presently, the lowest metrical level, the tatum, was the eighth notes of a bar and the algorithm had no mechanism for interpreting sixteenths. Whilst sixteenth notes are not always present in standard drum patterns, the syncopation in James Brown's 'Funky Drummer' displaces snare events by a sixteenth note. Sixteenth hi-hat

patterns are also found in rock music and have been particularly favoured by British Indie bands in recent years. These events would otherwise be interpreted as intended to happen *on* the beat and result in parameter re-estimation and possibly synchronisation, causing a rapid tempo variation.

The mechanism implemented to correctly interpret sixteenths is dependent upon the accuracy of the system. If the synchronisation window narrows, so that the standard deviation is less than a third of the interval between successive eighth notes (approximately 80msec at 120 BPM), then a notch is created for sixteenth events. An event within this region will be ignored by the system, which relies only upon eighth and quarter notes to update tempo and phase estimation. These notches can be seen in the accuracy function, shown in Figure 5.1.

5.1.2 The Layer Function

As previously introduced in Chapter 2, Lerdahl and Jackendorff's Generative Theory of Tonal Music characterises rhythm as a succession of strong and weak beats. They describe a set of rules to construct a hierarchy of metrical levels, so that a strong beat at one level is a strong beat at any lower level. B-Keeper defines each beat location as belonging to one of four 'layers', which correspond to Lerdahl and Jackendorff's metrical levels. The equivalent layer for the B-Keeper system is the number of dots below the corresponding bar position in Figure 2.1 of the GTTM's metrical structure shown in Chapter 2. B-Keeper also uses eighth notes between the beats, which are counted at the higher *tactus* level. Thus, if a bar is divided into eighth-notes, then the *layer* of a beat is determined by its position in terms of eighth-note intervals from the first beat of the bar.

Definition: The layer, l, is the maximum integer, n, such that the number of eighth-notes, q, from the first beat of the bar is zero modulo 2^n.

$$l = \max\{n \in \mathbb{N} : q \equiv 0 \bmod 2^n\} \tag{5.1}$$

Hence, the 'one' is layer 3, the 'two' and 'four' are layer 1 and the 'three' is layer 'two'. Events on the other eighth notes are layer 0.

The function works by storing which 'layer' the algorithm is currently 'on'. It will still synchronise if a beat of lower layer is observed which is more accurate than recent events at the higher layer, but will only change the model parameters for beats at the higher layer and above. In particular, it will not expand the window or lower the threshold unless a beat of the current strong layer is missed, when it drops down a layer and the window widens. The effect of the function is that the algorithm is 'looking' for strong beats at the current metrical level. It will maintain its tempo and phase through any amount of syncopation, so long as it can successfully find the strong beat beyond. This function allows the beat tracker to respond appropriately during fills and parts featuring expressive timing, and to make an interpretative response to each incoming beat on the basis of its bar position and recent playing history.

5.2 Case Study B : Free Improvisation

James Sedwards is a London-based guitarist in Nought, a instrumental experimental rock band. He collaborated with drummer Jeremy Doulton and Nought's original drummer, Al Pickard, to help evaluate B-Keeper's performance on improvised music. In order to do so, the system was synchronised by the supervisor (the author), and James Sedwards was free to loop sections of guitar and bass. Video of their performances is visible at the B-Keeper YouTube channel [1] and on the B-Keeper website[2]. Pickard's drumming is rhythmically complex and the duo have a long history of collaborative playing. He describes an interesting scenario that arose during the interactive process:

> "It stuck more to me. The only times it was thrown was when I played stuff I thought might throw it - that is too ambiguous for it to understand possibly. There's a certain scenario where I'm trying to throw it and maybe my instinct is re-adjusted to it and we're chasing each other. I'm so used to playing to a fixed

[1] `http://www.youtube.com/bkeepersystem/` as viewed 14th April 2009
[2] `http://www.b-keeper.org/` as viewed 14th April 2009

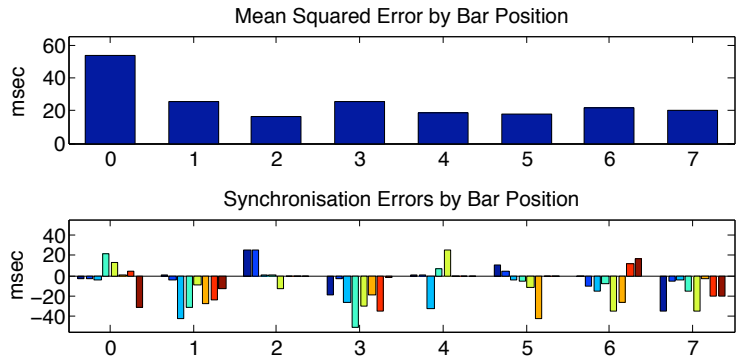

Figure 5.2: Errors from improvisation recorded with Al Pickard and James Sedwards.

> loop. I'll be playing and used to that. I'll shift the beat round, but [here] if that causes a change, then it starts to change and I'll adjust to it. This is loose playing, stuff that would confuse me. With James, we'll flip it on its head and the audience goes 'where are we?' "

5.2.1 Swing

The phenomenon of swing was discussed in Chapter 2. Swing and expressive timing are differentiated from performance error only by a regularity of occurrence. A beat that is 'swung', features a consistent uneven division of each pair of eighth notes which constitute the beat or tactus. A 'classic' swing is often conceived as having a triplet feel, whereby the first note is twice as long as the second and the corresponding ratio is 2:1. However, in practice, the ratio between the notes exhibits considerable variation and tends to decrease with tempo [Fri99]. A 'funky groove' will utilise a ratio little over 1:1, so that the first eighth note might only be 55 to 60% of the tactus interval.

The layer function enables beat tracking of swing patterns, since the swung events are likely to be those on lower metrical levels. We tested the system with looped recordings, made as demonstrations by Adam Betts when interviewed, and the resulting beat tracking is very successful. The quantity of swing for each event is represented in the graphics panel, shown

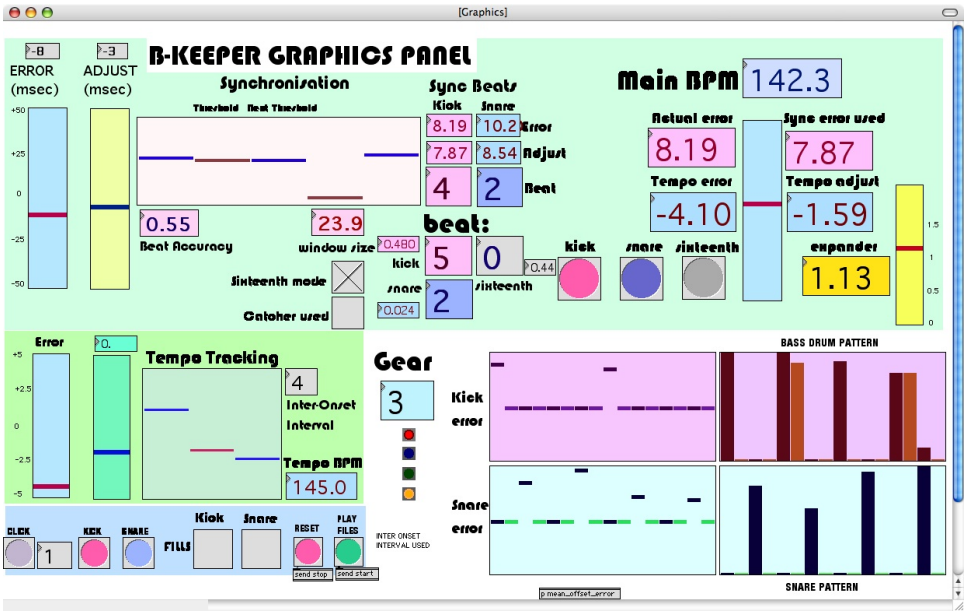

Figure 5.3: B-Keeper's graphics panel which displays statistical data to the supervisor or drummer.

in Figure 5.3. The provision for swing could be extended for jazz music by explicitly finding the constant pulse and discriminating between events which are swung and those which are on the pulse. Whilst B-Keeper can interpret swing on the weak beats, if strong metrical beats are swung, then this could cause difficulties. A graphics window, shown in Figure 5.3, displays the beats of the bar, performance errors, swing and the bass and snare drum patterns played.

5.3 Further Quantitative Evaluation

By examining the output of the algorithm, the adjustment of system parameters and statistical data such as recorded errors, we can check that the algorithm behaves the way we would want under different circumstances. In software engineering, tests may be carried out to ensure that requirements are met and stress tests may be carried out in more problematic situations than we envisage taking place.

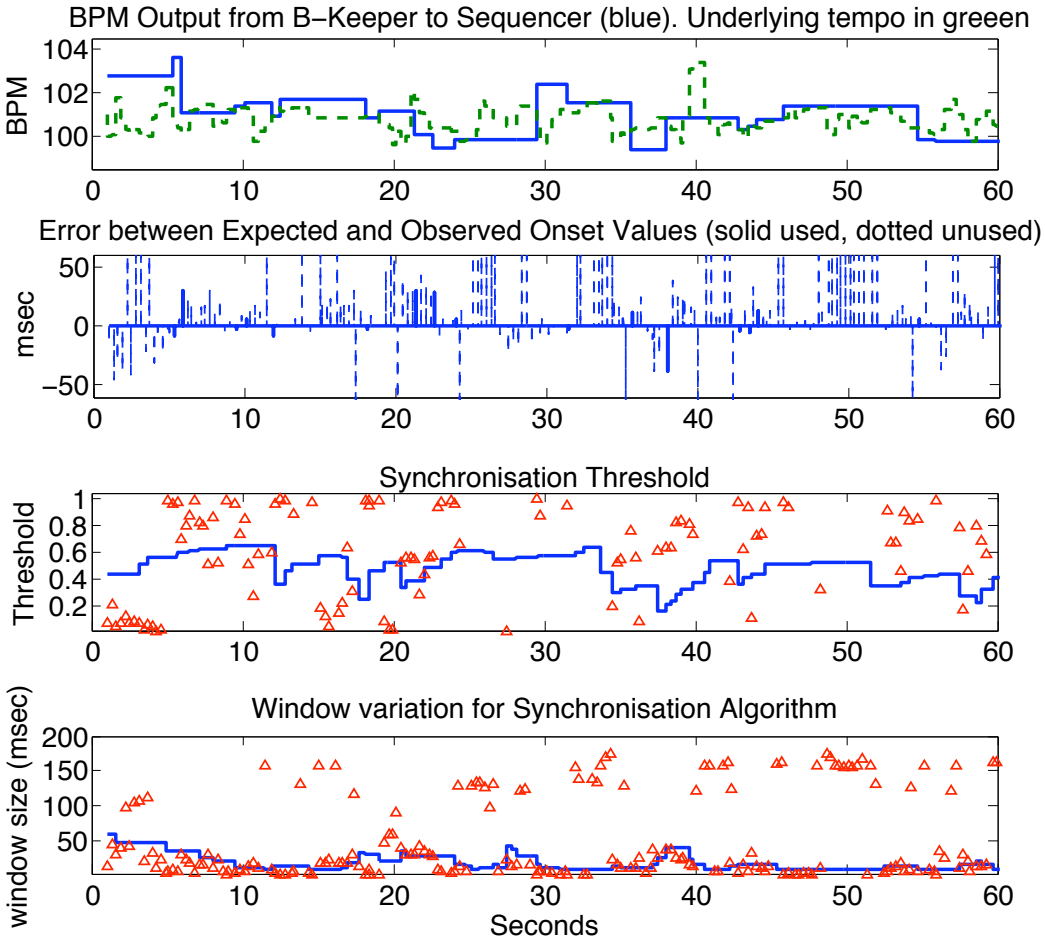

Figure 5.4: Response of the algorithm to the looped drum solo from James Brown's 'Funky Drummer'.

5.3.1 Analysis of James Brown's Funky Drummer

In order to test the beat tracker on a definitive example of syncopation, we looped the drum solo from James Brown's Funky Drummer. Freeman's [FL02] analysis by hand discovered expressive timing on the snare events, which were regularly displaced by up to 4.8% of the beat period.

The tempo output, shown in Figure 5.4, only varies between 100 and 102 BPM and thus is relatively steady. B-Keeper ignores many of the syncopated events, shown as dotted errors in the second graph. The synchronisation window, visible as the blue line in the bottom graph, is maintained below 40msec. The overall synchronisation errors are shown in Figure 5.5

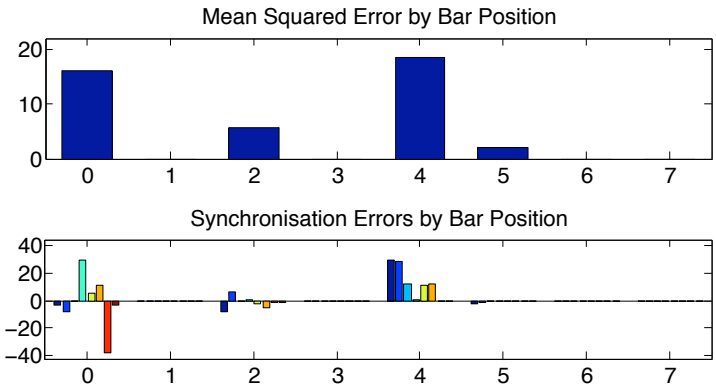

Figure 5.5: Mean squared errors for synchronisation to 'Funky Drummer'.

and these are within our desired perceptual threshold of 30msec.

5.3.2 Gradual Acceleration

In order to check the B-Keeper's response to a speed-up, drummer David Nock played to a click track at 120 BPM, heard in his headphones, which was subsequently sped up to 150 BPM over the course of 30 seconds. This maintained the higher constant tempo before returning to 120 BPM over the same interval. The algorithm's response to his playing is shown in Figure 5.6. The system reacts smoothly, increasing the window and lowering the threshold to accommodate the change in tempo. Due to the rapid rise in tempo, there is a latency between the speed up beginning at 35 seconds and the stabilisation at 48 seconds, when the error between algorithm and drummer is within the perceptual threshold. This is due to the time required for the algorithm's parameters to adapt sufficiently to compensate, shown in the third and fourth figures. In a performance, a speed up of this kind would be relatively rare.

5.3.3 Silent Accompaniment

One clear test that the algorithm is following the drummer is by not providing any audio feedback. In this case the system is functioning *reactively* to the drummer as opposed to interacting, but this demonstrates that the system can follow a metrical structure without performer compensation.

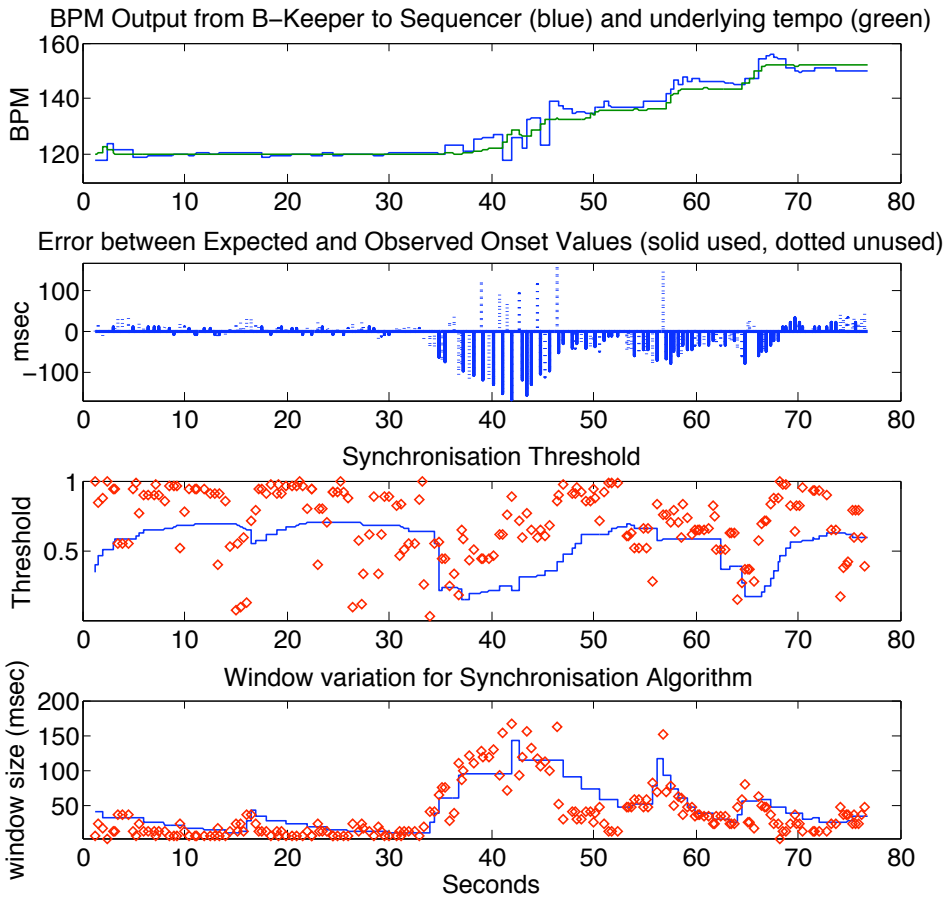

Figure 5.6: Output from B-Keeper with David Nock playing a regular drum pattern to a click track which speeds up incrementally from 120 BPM to 150 BPM over the course of 30 seconds. The accuracy result and errors recorded by the system are shown as diamonds in the third and fourth plots respectively.

During development, audio files of recorded drums were used to test the system, equivalent to silent testing, which aided the iterative development of the algorithm and highlighted potential difficulties.

During an evaluation session we examined the difference between interactive accompaniment and purely reactive (silent) accompaniment. During this test, B-Keeper successfully tracked the drums when no audio feedback was provided to the drummer, indicating that it is capable of reliable beat tracking. Interestingly, the errors, shown in Figure 5.7, recorded when feedback is given by the system (interactive) differed from those when the system tracked the drummer without providing auditory accompaniment.

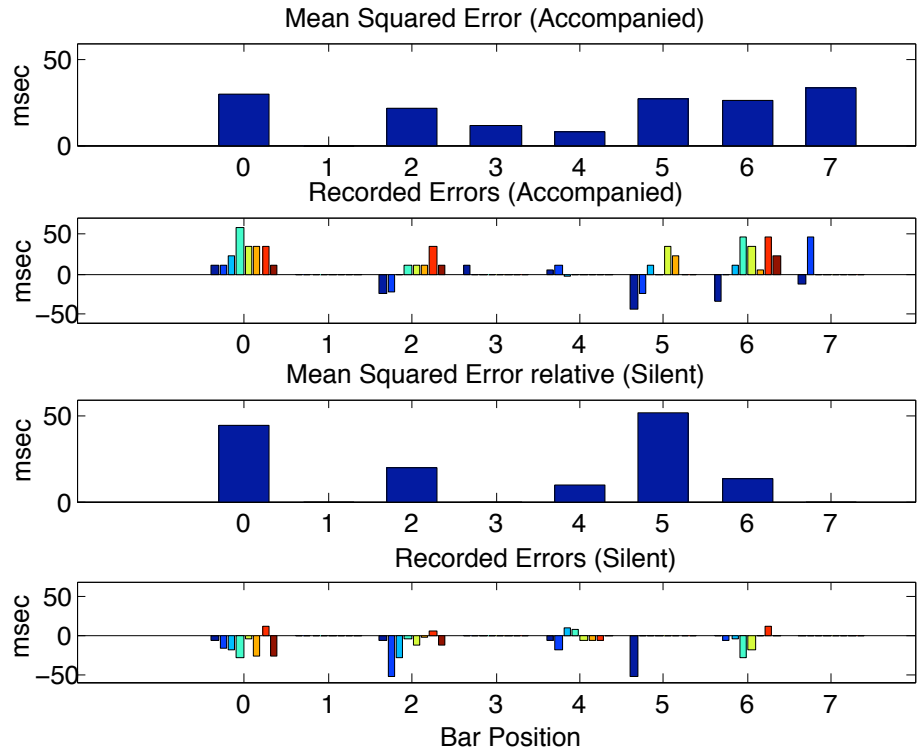

Figure 5.7: Errors recorded by David Nock with dance-rock piece. The top two figures show the errors recorded when accompanied by music from speakers. The bottom two figures show the results when playing the same drum pattern but without hearing the song. The errors are inverted here so +10ms means 10ms *early*.

In the latter case, the errors are still close to zero, as we would hope for a reactive system, but the placement of the drummer with respect to the click track has changed. It appears that the drums now are 'behind the beat' as opposed to 'ahead of the beat' when the accompaniment is audible to the drummer. This is consistent with David Nock's placement of beats when recording in the studio with an accompanying click track at a fixed tempo and may relate to the previously discussed phenomenon of negative asynchrony [Asc02]. A control is provided on the B-Keeper interface so that the accompaniment can be offset relative to the interpreted beat if required.

5.3.4 Other Instrumentation

Initial tests using the B-Keeper on acoustic guitar suggest that it might be possible to process other musical information when there is no signal from the drums. B-Keeper's methodology makes it suitable for adaptation for other instruments. By using a very simple energy-based onset detector and suitable settings on the B-Keeper, a percussive backing was provided to an acoustic guitar input to the system.

5.4 Summary

In this chapter, we have presented new modifications to the beat tracking algorithm as a result of extensive testing with live drummers and with recorded audio files. By modifying the system's parameters differently depending upon information from top-down processing, the system improves its negotiation of complex syncopation, tempo change and expressive timing. We have also designed a new graphical interface to present this information to the drummer or supervisor when using the system. This interface displays performance information relating the drum pattern played on the kick and snare, error information relative to the predicted beat, swing information and shows the real-time adaptation of system parameters such as thresholds and window sizes in response to the musical signal.

Chapter 6

Real-Time Multi-Pitch Tracking

Having developed a system for beat tracking with drums, we will now consider the case of semi-percussive instruments such as guitar and piano. Beat tracking systems could function well on such signals to provide tempo estimates, but for automatic accompaniment we still require accurate phase determination. Previous automatic accompaniment systems [Dan84] [Ver84] [Rap01] [Con08a] have made use of pitched notes to match an accompaniment to a score. The detection and matching of salient pitched events might be used to synchronise live and recorded audio sources, thereby requiring real-time pitch tracking. Since we require this for instruments that are not monophonic, we will first investigate real-time multi-pitch tracking which outputs a MIDI representation of an audio input. An alignment algorithm that synchronises two audio sources by matching the resulting MIDI streams from the pitch tracker is presented in the appendix. In addition, a polyphonic pitch tracker would useful as input to generative systems or in the creation of real-time musical controllers.

6.1 Introduction

Polyphonic or multiple pitch-tracking is a difficult problem in signal processing. Much of the existing work in multi-pitch tracking has been in the field of Music Information Retrieval which takes place offline on large data

sets. Previous research into pitch tracking for interactive music has highlighted the importance of minimal latency and accuracy within noisy conditions [dlCMS01]. Since our algorithm is employed for real-time audio-to-MIDI conversion within a performance system, we require fast detection of notes and low computation time.

We will first look at some previous approaches to the problem before presenting a real-time algorithm for audio-to-MIDI conversion.

6.2 Pitch tracking techniques

A method for multiple frequency estimation by the summing of partial amplitudes within the frequency domain was presented by Klapuri [Kla03] [Kla06], who makes use of an iterative procedure to subsequently subtract partials within a pitch detection algorithm. The salience $s(\tau)$ of a period candidate is given by the equation

$$s(\tau) = \sum_{m=1}^{M} g(\tau, m)|Y(f_{\tau,m})| \tag{6.1}$$

where $f_{\tau,m} = mf_s/\tau$ is the frequency of the m^{th} harmonic partial of a F_0 candidate f_s/τ, f_s is the sampling rate, and $g(\tau, m)$ defines the weight of partial m in the sum. $Y(f)$ is the short time Fourier transform of the signal. In practice, the discrete Fourier transform is used. The signal is pre-processed using spectral whitening in which regions of greater spectral energy are 'flattened' to suppress timbral information, allowing the same 'pattern' of partials to be applied to all sounds. Since any fundamental frequency f_0 produces several peaks in the salience function $s(\tau)$ at periods corresponding to its partials, an iterative estimation and cancellation scheme is employed. Iteration ceases when the quantity

$$S(j) = \frac{\sum_{i=1}^{j} \hat{s}(\tau_i)}{j^{\gamma}} \tag{6.2}$$

no longer increases, where $\gamma = 0.70$ (found empirically) and $\hat{s}(\tau)$ is the approximated salience using discrete Fourier transform. With a framesize of 46ms, the error rate is under 10% for monophonic audio, and approximately 15%, 27% and 36% (errors estimated from graph) for polyphonies

of two, four and six notes respectively. The number of sounds in the mixture was provided to the estimator.

Pertusa [PI08] lists potential fundamental frequency candidates in order of the sum of their harmonic amplitudes. At least two partials must have amplitudes exceeding the threshold for any given fundamental to be detected. The salience of all possible chord combinations is calculated per frame and discontinuities of detection are reconciled. Short notes less than six frames (56ms) are discounted. The accuracy on the same data set of mixtures used for evaluation by Klapuri is 56.21%.

Zhou and Reiss [ZR08] make use of a complex resonator filter bank to produce a time-frequency energy spectrum. A combination of onset detection and multiple pitch estimation is used to transcribe polyphonic audio, with the first four partials contributing to the energy recorded at any given fundamental.

Algorithms for pitch detection need to account for inharmonicity in the signal, whereby the ratio between partial frequencies and the fundamental departs from perfect harmonicity. Wen and Sandler [WS05] explicitly learn calculate the inharmonicity present in partials in order to aid the pitch tracking process.

Approaches using non-negative matrix factorisation were originally introduced by Smaragdis and Brown [SB03] and Abdallah and Plumbley [AP04]. These attempt to learn a 'dictionary' of atomic sounds and, assuming sparsity, a signal can be decomposed as a linear sum of a small number of these auditory atoms. For music transcription, each element of the dictionary can model the combination of partials present in a pitched note so that the atoms correspond to the individual pitches. The unsupervised learning procedure, used to compute the dictionary, can be computationally intensive. Cont [Con06] [CDW07] has applied the method to real-time transcription by learning instrument templates. The resulting *transcribe*$\sim$ object has been used to generate input for a score following system. One problem confronting such algorithms is that they require static harmonic profiles, so that the 'objects' used to describe a note do not change if the ratio between partials varies as the note decays.

Existing real-time algorithms for pitch detection include *fiddle*$\sim$, a

Max/MSP object by Miller Puckette based on a Fourier transform which employs peak picking. Tristan Jehan [JS01] adapted the algorithm to analyse timbral qualities of a signal. In the time domain, Alan de Cheveigné's Yin [dCK02] is a widely-known algorithm which uses auto-correlation on the time-domain signal to calculate the most prominent frequency. However, both of these algorithms are designed for monophonic signals and they are not reliable enough to generate a MIDI transcription of audio from a polyphonic instrument.

6.3 Approach

We proceed from the assumption that a pitched note also causes peaks to occur in the spectrum at frequencies corresponding to partials of the fundamental frequency. We iteratively subtract these partials within the frequency domain in order to aid a real-time pitch detector. A learning method is employed to optimise the expected amplitudes of the partials of each detected note by continually updating the weights whenever a note is detected. In addition, we model the variations within the amplitude and summed partial amplitudes of detected notes. The weightings for each partial derived from observations are used within the decision-making process that triggers a MIDI note-on message.

Implementation

Our algorithm has been implemented in Java within a Max/MSP patch and in doing so, we made use of the *fiddle*~ object by Miller Puckette [PAZ98], in the pre-processing stage. Ordinarily, *fiddle*~ provides its own fundamental frequency estimation, but it also gives the raw data of the top N frequencies from the peak picking process and their respective amplitudes above a suitable threshold. Since *fiddle*~ has been optimised for fast processing within a real-time environment, it is well-suited to providing an efficient FFT and noise reduction process used to provide the data for our partial-removal system. We use a frame of 2048 samples with a hop-size of 1024, so that our detection of notes is a fast as possible whilst still detecting pitches as low as 80Hz.

6.4 Method

Find peak frequencies in the spectrum

The 'uncooked' output of fiddle~ provides a real-time list of the top N frequency peaks of the spectrum and their amplitudes, where typically N is between 8 and 20 for polyphonic audio. The system is capable of better polyphony if N is higher.

Update the amplitudes

The algorithm continually tracks the amplitude of bins in the frequency domain delineated by their corresponding MIDI note. First we calculate the MIDI note for each of the incoming peak frequencies and update its respective amplitude. We accept only a small number, under twenty, peak frequencies from the spectrum. All other amplitudes are decreased by 20% for each input frame (every 23 msec) to allow for errors if notes are accidentally skipped in this 'top 20' procedure. It is quite common in the duration of a note, for one of the peak frequencies to shift to an adjacent note in a frame and this prevents the original amplitude dropping to zero (triggering a note-off).

Track new and existing notes

We begin with the lowest note and work up the range of frequencies. For every note present in the incoming peak frequencies list and every note already playing, we calculate the 'power' of the note, $P(m)$, by summing the product of the amplitude of the MIDI bin corresponding to the respective partials with a weighting matrix, $W_m(k)$, for that note. This is given by

$$P(m) = \sum_{k=1}^{L} W_m(k) A(m + h[k]) \tag{6.3}$$

where A(m) is the amplitude of MIDI note m, L is the number of partials summed (we chose $L = 6$), k is the partial number (the note's frequency as an integer multiple of the fundamental frequency), $h[k]$ is the interval in semitones between frequencies f_0 and kf_0, and $W_m(k)$ is the weight

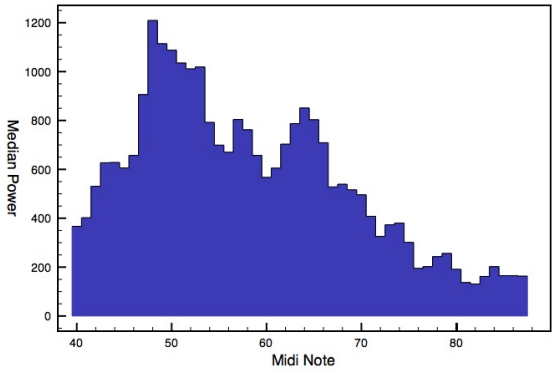

Figure 6.1: Median power for triggered notes over the range of piano notes. The power of played notes varies dramatically with pitch so that learning the median value for triggering plays an important role.

vector, derived from the observed signal, of the amplitude of the k^{th} partial relative to the amplitude of the fundamental, specific for each individual note in the spectrum.

Notes On and Notes Off

For currently playing notes, we look for a note-off event:
If $P(m) < \theta_{-}.\bar{P}(m)$, then output a MIDI note-off event for pitch m,
where θ_{-} is a threshold parameter and $\bar{P}(m)$ is an estimate of the median power of a positively detected note, dynamically calculated from triggered MIDI note-on events. Figure 6.1 shows how this quantity varies over the range of played notes.

For non-playing notes, we calculate the change in power as a ratio between the current frame and the previous frame.

$$r(m) = \frac{P_t(m)}{P_{t-1}(m)} \tag{6.4}$$

For a note-on event, we require that both the power of the note (the sum of partials) *and* the amplitude (the fundamental) exceed suitable thresholds. Thus, we check that the MIDI note has at least one partial note $m + h[k]$ that is one of the top N peaks present in the incoming data from fiddle~, such that $k <= 4$. If the only partial present is $k = 3$ (19 semitones above the fundamental), then we require that $(m + 7)$, the fifth, is not also a

peak. Then, the rule for a note-on is as follows:
If $P(m) > \theta_+.\bar{P}(m)$ and $r(m) > \theta_r$ and $A(m) > \theta_a.\bar{A}(m)$, then output a MIDI note-on for pitch m,
where θ_+, θ_a and θ_r are thresholds for the power, amplitude and ratio, $r(m)$, respectively.

This requirement ensures a significant measure of summed harmonic amplitudes and a significant increase in this measure since the last observed frame. In practice, values for the ratio threshold, θ_r, tend to be between 1.4 and 3, depending on the level of response required. The higher the ratio, the less likely the algorithm is to trigger a false positive.

In the case of a new note-on or if the current note was triggered within the last three frames, we adapt our weights $W(p_n)$ that are used in the summation process. Our current observation suggests:

$$W^*(k) = \frac{A(m + h[k])}{A(m)} \tag{6.5}$$

We track how many observations have been made in the past and adapt so $W(p_n)$ is the average of these and the new observation $W^*(p_n)$. We perform this update for all notes within 5 semitones of the played note since some notes are played less frequently, yet we can reasonably assume that the tone and timbre with respect to partial weightings is approximately the same as the surrounding notes. By including notes close to our observed note, we adapt the weights more quickly to a useful approximation.

We also update our estimate for the median of the amplitude and power of a note out at that MIDI pitch using an exponential moving average:

$$\bar{A}(m) = (1 - \alpha).\bar{A}(m) + \alpha.A(m) \tag{6.6}$$

where α (typically 0.2) defines the response of the median estimate to new data.

Partial Subtraction

Having evaluated the current note's strength, if the note is either playing or a new note, then we subtract from the amplitudes of its partials higher in the frequency range. High frequencies will have considerable amplitude

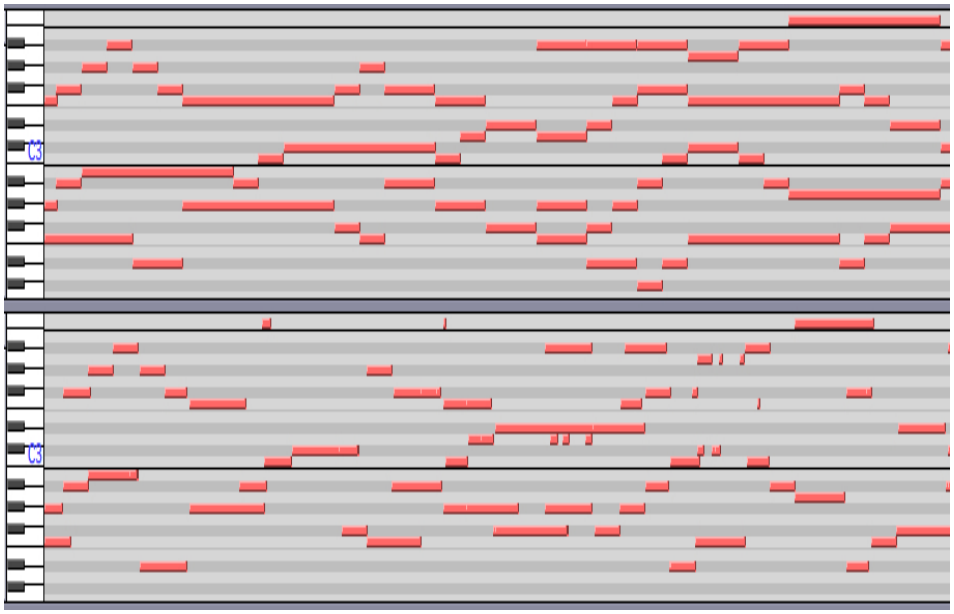

Figure 6.2: Ground-truth MIDI from Bach's 'Well-Tempered Clavier' (top) and the MIDI output from the corresponding synthesized audio as input to the pitch-tracker (bottom).

due to this lower fundamental, so the subtraction process helps to prevent false positives from partials. Hence, we use the following update rule:

$$A(m + h[k]) = A(m + h[k]) - W_m(k)A(m) \tag{6.7}$$

for $1 \leq k \leq L$. We aim to optimise these weights by introducing some feedback at this stage. If the subtraction process results in $A(k)$ becoming less than zero, then we decrease $W_m(k)$. If it is greater than zero then we increase the weight. Hence, all playing notes function to optimise the weighting function for the note and its associated neighbouring pitches.

There is an assumption here that for the majority of notes the instrument is *relatively monophonic*. The weights are adjusted on the basis that if a fundamental is playing, the partial is not also playing as part of a polyphonic chord. Whilst this may not be strictly true (as when an octave plays), we hope that it true for the most part, so that when an octave does play, the residual power in the first partial after the subtraction process should still be substantial enough to trigger the recognition of the octave note.

Piece	Correct	False Positive	Number of notes
WTC1f	80.0	31.3	1075
WTC1p	71.0	39.3	833
WTC2f	76.4	36.8	647
WTC2p	78.4	27.3	1408
WTC8f	79.0	41.0	1014

Table 6.1: Detection Rates against synthesized harpsichord audio from Bach's Well-Tempered Clavier.

6.5 Evaluation

We have used the algorithm in live performances on an acoustic guitar, using it to create a texture of synthesized sounds behind the guitar. By filtering notes to an appropriate scale, we can help to avoid dissonance from false detections. Experimentation with a MIDI-triggered electric piano sound suggests that there is a detection latency between 60 and 90 msec. This is still quite considerable for use within a live context when fast passages are played. By comparison, Miller Puckette's *bonk*~ onset detector has a latency of approximately 10 to 30 msec for the same notes. However, the onset detector is able to make use of a frame-size of 256 samples (or 5.8 msec), whereas for the Fourier analysis involved in adequate pitch detection, we require a frame-size of 2048 samples (or 46 msec).

When used within performance, this provides good subjective results. To our knowledge, there is no existing Max/MSP polyphonic real-time object available for direct comparison. The *fiddle*~ and *yin*~ objects are monophonic and were not designed for polyphonic pitch detection, and their use in this context gives subjectively poor results in comparison. We would like to provide an objective measure of success within a performance application. However, we can so far only compare with offline systems.

On this, we tested the tracker on several synthesized harpsichord recordings of Bach's Well-Tempered Clavier. By sending MIDI files to a Yamaha Stage Piano and testing the pitch-tracker on the corresponding synthesized audio, we can simulate the task of audio-to-MIDI conversion

	Piece	Detected (%)	False Positive	Number of notes	Accuracy (%)
Synthetic					
	BWV828	46.0	38.0	496	35.9
	Humoresque	42.1	49.6	545	27.8
	Sonata no.15	61.1	22.4	651	52.0
Real					
	BWV810	50.1	43.7	652	39.2
	Nocturne no.2	38.1	37.6	252	21.2
	Entertainer	45.1	42.7	567	33.8

Table 6.2: Detection Rates against the piano data set used to test Sonic.

for a polyphonic instrument, whilst having ground-truth of the notes actually triggered.

A representation of the MIDI ground truth and the corresponding output from the detector is shown in Table 6.5. The average latency measured between 70 and 90 msec. The percussive, distinctive nature of the harpsichord sound seems to be an optimal input for the pitch-tracker resulting in high performance statistics of approximately 80% correct detections. The precision currently appears to be comparable with some offline trackers. The MIREX 2007 [MMDK07] competition results rate offline trackers with a precision of between approximately 40 and 70%. The equivalent precision for our real-time pitch-tracker here would be over 50%, but it is important to note that the MIREX competition uses a wide database of varied sounds and hence the result on the Bach pieces may be artificially high.

Marolt [Mar05] has developed an offline pitch-tracker, Sonic, specialised for piano input, which uses adaptive oscillators and neural networks. We also tested the tracker on excerpts from his data set using a selection of three synthesized audio samples and three performances with a real piano.

The synthesized pieces were: J. S. Bach, Partita no. 4, BWV828, ; A. Dvorak, Humouresque no. 7 op. 101, ; W. A. Mozart, Sonata no. 15 in C major, K. 545, 3. mvm.

The real pieces were: J. S. Bach, English Suite no. 5, BWV810; F. Chopin,

Nocturne no.2, Op. 9/2; S. Joplin, The Entertainer.

The results are shown in Table 6.5. Sonic obtains a success rate of approximately 90 % on this data set, with a false detection rate of approximately 9%, whereas our real-time tracker only succeeded in detecting between 40 and 50% of the notes with considerably less precision (approximately 40% false positives). A significant proportion of the false positive rate is due to high frequency content from harmonics present within the original signal, which are more tolerable in a live context than inharmonic and lower frequency errors. The accuracy measure, first proposed by Dixon [Dix00], is given by:

$$\text{Acc} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}} \tag{6.8}$$

where TP is the number of true positives (correctly identified notes) and FP and FN are the number of false positives and false negatives respectively. An alternative frame-based evaluation metric is given by Poliner and Ellis [PE07].

Our proposed system gives an overall accuracy result of 35.0% over the six pieces in offline tests. The average polyphony on these pieces varied from 2.7 to 4.4, which partially explains the low correct detection rate since in these tests we used only the top eight peaks from fiddle~ as input. Since we required both fundamentals and partials to be present for a detected note, this limits the potential polyphony that our system can accommodate. Marolt's results correspond to an average accuracy measure of 77.5% over six pieces used for evaluation, five of which were used in our evaluation.

Although our figure is significantly worse than specialised systems designed for offline polyphonic transcription tasks, the transcription is made in real-time and our subjective observations are that it is very successful in a live performance context. This raises the issue of how we can perform an evaluation that fairly reflects success in such performances. In Chapter 4, we used subjective evaluation to assess the success of B-Keeper and this is a potential direction for future work. Whilst the accuracy is not as high as we would like for a multi-pitch tracking system, it is also useful as input for an audio alignment algorithm which is describe in the Appendix.

6.5.1 Potential Improvements

There are considerably many false positives in the higher registers, possibly corresponding to higher partials whose frequencies are of integer ratios higher than six relative to the fundamental pitch. Our algorithm has been modelled explicitly on the kind of signal we expect to observe when a note plays. We expect a note with a given fundamental pitch to cause the observation of pitches corresponding to the higher partials of that fundamental. These are tracked in the observation process, firstly contributing to the positive detection of the fundamental pitch as a note, and secondly, they are removed from the observed pitch classes, so as to prevent false positives. However, only the first few harmonics have been accounted for and there are several false positives in the higher registers, such as at 34 semitones from the fundamental, corresponding to a frequency seven times the fundamental, which are not subtracted at present. In addition, it may be the case that there are other pitches which are regularly observed, whose frequency does not have an integer ratio to a fundamental present. These can be regarded as noisy observations. Thus, it may be possible to subtract these noisy observations in a similar fashion to the partials.

At present, we have assumed that a note creates minimal noise in other bins than the partials. This simplification could be rectified by allowing the note to create noise in all bins and learning the parameters in real-time. It is strongly related to the problem of sparse coding [AP04], since we would be using a non-negative matrix A, where $a_{i,j}$ represents the contribution to MIDI bin i from a played note j. The array of amplitudes We aim to investigate a new real-time implementation of the algorithm with this new formulation.

6.6 Summary

We have presented a system for real-time polyphonic pitch tracking for live performance applications, based on iterative subtraction of estimated partial amplitudes from the frequency representation. Our approach uses

a fast deductive procedure based on the existence of partials for any given note. By continually updating estimates for the weight of the partials relative to the fundamental, the median values for the amplitude and power of all notes, our algorithm is capable of performing moderately well on databases designed for offline multiple pitch-tracking algorithms. Although the algorithm does not perform as well in objective tests as other algorithms designed for offline use, our approach does give subjectively high success in a live performance application.

Whilst this algorithm might be used to control MIDI devices from conventional instruments, it can also provide harmonic information that may be useful for performance systems. In the appendix, we present initial work on an algorithm that attempts to synchronise recorded audio with a live rendition. In this case, MIDI information resulting from the pitch tracking algorithm is used to match between the two streams rather than using rhythmic information. The pitch tracker could also be used in interactive performance systems which make use of harmonic information to generate a musical response.

Chapter 7

Conclusion

In this thesis, we have designed and evaluated a drum-based automatic accompaniment system for rock bands and looked at other approaches that may contribute towards the design of interactive music systems. Both B-Keeper and the tracking algorithm presented in the appendix make use of a design architecture in which parameters are used to describe the aspects of the system's behaviour. Each system processes temporal information about high-level musical events (onset times in the case of B-Keeper and MIDI note events in the case of the tracking algorithm) which result in a tempo output to control the accompaniment and an adjustment of the system parameters to control behaviour. The pitch tracking technique presented in Chapter 6 provides harmonic information that may be of use, either for alignment of an accompaniment system or when determining the output of a generative system.

We have also made a contribution to evaluation studies for interactive musical systems. The B-Keeper system has been evaluated using both offline tests and through the musical 'Turing Test' which involved many drummers. The results have been very positive, suggesting that it behaves in a 'human-like' manner. There is a link here to research in cognitive psychology, where experiments by Repp and others have led to the proposal of a two-process model for human beat detection, in which phase recovery is near-instantaneous and sub-conscious, whereas adaptation of tempo is slower and involves recognition. B-Keeper's main method to maintain synchronisation is via fast adaptation of the accompaniment's tempo to rectify the observed phase differences, whilst adjustments to the underlying tempo are slower to occur. The success of B-Keeper in experimental

evaluation trials suggests that the two-process model is a viable when implemented as a performance system for drums.

7.1 Thesis Contributions

We now summarise the main contributions of this thesis.

Drum Tracking

We present a novel method for drum tracking, incorporating top down metrical information into bottom up event based processing. This brings about an interpretative schema in real-time which simulates musical understanding. The drum tracking system successfully follows the meter of difficult drum passages, even when audio feedback is not provided to the drummer. This level of synchrony is a major contribution towards a full system for performance following.

Musical Turing Test

We introduce the musical Turing Test for the purposes of evaluating musical systems where their role is one which could be carried out by a human. In other fields of research, 'discrimination tests' which make clear analogies to the 'Turing Test' have been used to evaluate algorithms, however, by re-introducing the element of interaction present in the original test, the formulation is closer to Turing's original conception.

Multi-pitch tracking

We present a new real-time method for transcribing polyphonic audio to MIDI which models the presence of partials of higher frequency within the spectrum. Few real-time polyphonic pitch tracking algorithms currently exist. Whilst its performance does not rival offline algorithms for detection rate, the information provided could be valuable for use within interactive performance systems.

7.2 Future Work

The problem of tempo and phase induction is potentially separate from that of how to maintain synchronisation. We have focused on the latter problem which involves categorising new events and making use of key information to update the hypothesis. Whilst this strategy has appeared successful, it appears that the problem of synchronisation is more complex than may be first supposed. We now describe some potential areas for future research that have emerged from the work presented in this thesis.

Drum Tracking

B-Keeper is a beat tracking system that synchronises with a drummer and is stable to fills, syncopation and tempo microchanges. The system is available for download [1] and a version will be developed for the Max4Live environment which integrates Max/MSP into Ableton Live.

At present, the system does not analyse audio input to detect tempo, phase and bar boundary, but accepts a specific starting condition (either by cue or through user input). A complete system for drums might analyse audio to generate an approximate tempo hypothesis. For example, Davies and Plumbley's [DP07] work on tempo detection could be integrated with B-Keeper to form a more complete system for drum tracking whereby their algorithm could provide an initial tempo and phase estimation. Once a stable estimate has been reached, these values could initialise the estimates of B-Keeper system and trigger the accompaniment, thus allowing a drummer to begin playing and the two beat trackers work in conjunction with no prior information.

A robust system would be capable of handling more extreme events. There is provision for a supervisor to intervene to recover from error, but an automatic recovery process which could analyse the structure of a piece *across* the temporal domain would be a significant advantage. In his book 'Sweet Anticipation', David Huron [Hur06] extends Meyer's [Mey56] theory of meaning through expectation and describes the ITPRA theory,

[1] `http://www.b-keeper.org`

motivated by the proposition that musical expectation involves five functionally different psychological systems: imagination, tension, prediction, reaction and appraisal. Whilst *prediction* is an important psychological process in forming our response to music, retrospection also plays an important role through *appraisal*, whereby an event continues to be analysed once further information has been received. Our system uses prediction to react, but does not yet have any provision for short-term retrospective analysis to inhibit or verify that response which might help to discriminate between ambiguous events and enable stability in more extreme cases where the rhythm is more variable.

Pitch Tracking

Our assumption that a pitched note can be decomposed into a simple sum of sinusoid partials is an idealisation. In practice, the Fourier spectrum resulting from a pitched note also contains energy in other bins, such as those adjacent to all bins corresponding to partials of the given note. Therefore, it makes sense to extend our approach by formulating the problem as one of sparse decomposition using a non-negative matrix factorisation, whereby a pitched note would cause a proportion of energy in all bins rather than just the partials. Our aim will be to learn the sparse decomposition for each pitched note in real-time and use this to optimise the audio-to-MIDI triggering process. There is also the possibility that timbral features could be analysed to extend the description beyond that offered by MIDI representation.

Performance Following

There is potential to develop performance systems for a variety of instruments and for combinations of those instruments. At present, the tempo tracking algorithm in the Appendix measures only the relative tempos by aligning MIDI information provided by the pitch tracking algorithm. We would like to extend this algorithm by incorporating tempo estimation on both streams of audio, so that salient features can be used to match the phase of the signals whilst tempo could be estimated from both harmonic and rhythmic features. This approach would improve reliability by

interpreting live performances with harmonic instruments in a top-down manner analogous to the rhythmic interpretation provided by B-Keeper for drums. Further investigation into the representation of musical signals might help this matching procedure. Both the event-based onset detection used for drums and the use of MIDI representation for harmonic signals have the benefit of simplicity, but they are less suitable for the representation of signals from instruments such as guitar or strings that do not follow the keyboard paradigm.

7.3 Final Words

An important outcome of this thesis has been the necessity for phase adaptation within tempo tracking systems for rock and pop music. Our work in beat tracking suggests that the mechanism for synchronisation needs to be highly accurate with respect to phase if the system is to be useful for real-time accompaniment. We have also made use of a design strategy in which the system changes its parameters as a result of incoming information. The ability to recover from error and use multiple modes of behaviour which can adapt to suit the musical context are critical for autonomous systems.

Reflecting upon the work emerging from music psychology, there are still several problems confronting real-time beat tracking system for all types of instruments. We have mainly focused on the task of maintaining synchronisation, but there are other problems for real-time systems that relate to tempo and phase induction: when to start a song, how to recover from mistakes, how to estimate tempo, phase and bar boundaries. If an algorithm could gauge the reliability of its estimate, this would be valuable when integrated into a larger performance system.

The unification of algorithms specialised for different instruments would enable many new creative possibilities. An intelligent system capable of following the musical structure of a performance could provide an interactive accompaniment, generate a musical response, and, by extension, could control visual aspects of the show such as robotics, video images and lighting. Such a performance environment would respond both

sonically and visually to the artists. This is an exciting direction for new research at the boundary of art and science.

Appendix A

A Preliminary Algorithm for Audio Synchronisation

We are interested in the problem of automatic accompaniment for harmonic instruments such as guitar and drums. Whilst rhythmic information might allow beat trackers to estimate the tempo of such signals, the notes and chords played are the most useful information for phase alignment. The multi-pitch tracker described in the Chapter 6 has been intended for use as input to an audio synchronisation algorithm. We will now look at previous work in audio alignment and the related problem of score-following before describing our approach to the problem.

A.1 Tracksuit: An Algorithm for Audio Synchronisation

With the exception of Dixon's MATCH algorithm [Dix05], accompaniment systems have tended to focus on *scored music* where it is natural to formulate expectations on the basis of the next notes predicted by the score. Each performer imposes their own interpretation by varying the tempo and introducing expressive timing, but the same symbolic structure gives rise to our aural observations. Here we are aiming to provide accompaniment to music which has similarity of structure but no score.

There are different possible approaches. We could look to extract rhythm, tempo and timing information from the audio signal and use this to synchronise the accompaniment. Alternatively, we could look for salient events within the audio and seek to match these to their occurrence

in our accompaniment. The B-Keeper system used the metrical structure and tempo estimate, yet no prior knowledge of the performer's part to successfully synchronise drums. We will now describe a system developed which has only a *relative* tempo estimate and uses pitched note events to synchronise audio with an accompaniment. A pre-recorded version of the same piece effectively provides *prior information* about which musical events occur in the accompaniment, but this information is provided in real-time as synchronised audio rather than stored or learned from rehearsal.

A.1.1 Overview

Our motivation in developing the pitch tracker is to provide automatic accompaniment when this could not be done via beat tracking of drums, such as for a performance with a solo acoustic guitar or piano. We formulate the problem as one of matching a live audio stream to pre-recorded audio of the same part. We make the assumption that the pre-recorded version can function as a prior for the expectation of relative timing between acoustic events. Where this is an alternative take by the same musician, it is likely that there is a correspondence between variations in tempo and timing for the two takes.

We use the algorithm described in Chapter 6 to create a symbolic MIDI representation of incoming audio and perform alignment upon both two streams. Rather than estimate the tempo explicitly via beat tracking, we choose to estimate the *relative* tempo and alignment or phase difference through a sequence of matched events and thereby output a change in tempo for the accompaniment to maintain or improve synchronisation. Since pitch tracking is prone to error and the observations are potentially noisy, this method functions as a computationally efficient averaging process which uses the most accurate new observations to update the tempo. The system is predictive since the song position of the recorded audio stream can be either ahead of or behind the song position of the live audio stream.

We assume that the algorithm has an estimate for the tempo and

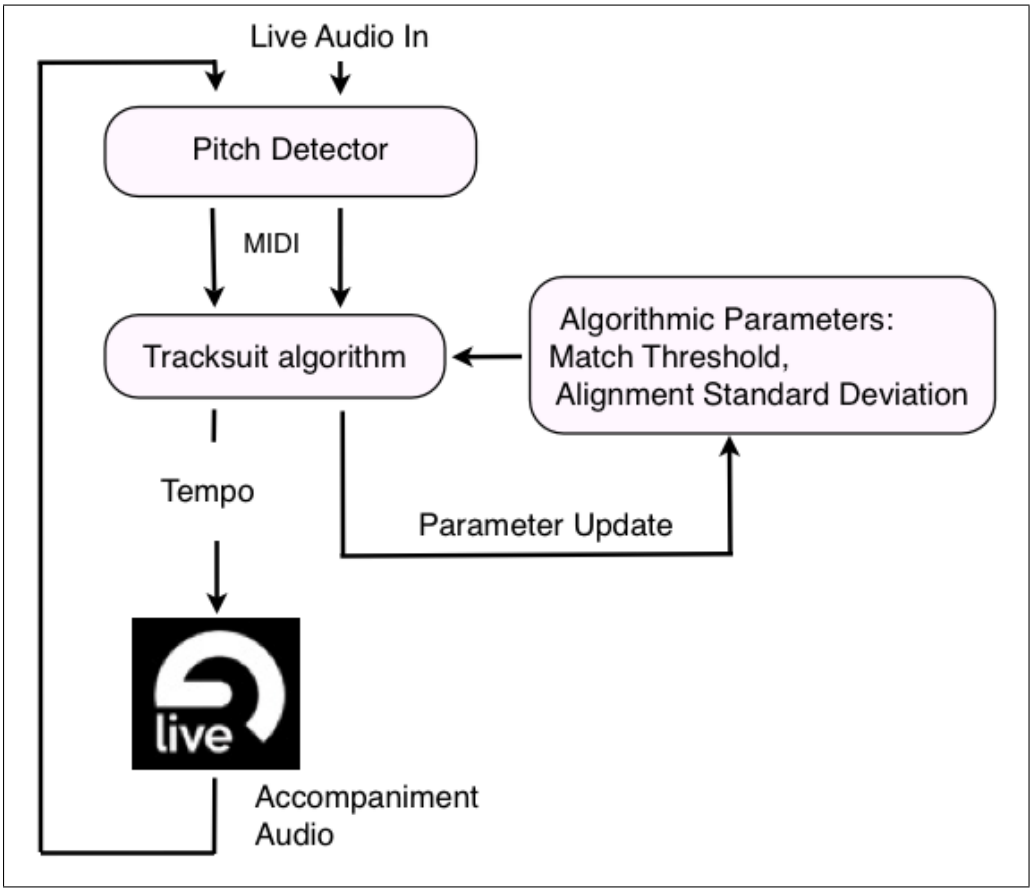

Figure A.1: Illustration showing the design of the algorithm. Relative tempo and parameter re-estimation take place as a result of new alignments made of the live and accompaniment MIDI streams.

phase that is *approximately correct*. This is satisfied initially by providing a starting tempo and MIDI value of the initial note that triggers the accompaniment. Such concessions would be appropriate in a performance system. Additional parameters determine the behaviour of the system and will be described below. Both the multi-pitch detection and the alignment algorithm are implemented as Java externals within Max/MSP. Ableton Live [1] is used as the audio sequencer to play the accompaniment audio since the time-stretching capabilities of the software allow adjustment of the tempo in real-time using two 'warp markers' at the beginning and end of the recorded audio file as shown in Figure A.2.

[1] `http://www.ableton.com`

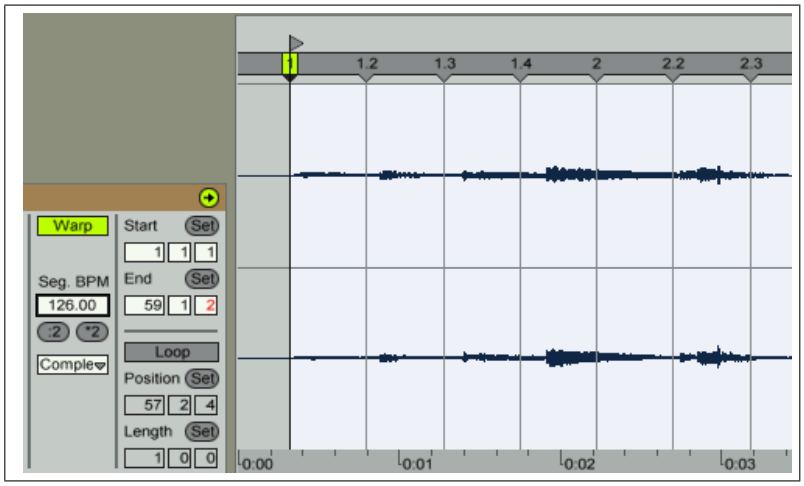

Figure A.2: Warp marker in Ableton Live placed at the beginning of the audio file. The tempo is set so that at 126 BPM it plays at normal speed.

A.1.2 Match Measure

The pitch tracking algorithm provides MIDI note-on and note-off data as input to the tracking procedure. We rate the match of a pair of notes according to the following criteria:

- Exact or partial Match, where one note is identical or corresponds to an octave of the other
- Surprise or novelty of the note measured against the recent past (an entropy-related quality)
- Volume
- Pitch: lower notes are considered more salient

Our inclusion of surprise or novelty is based on the need to make genuine matches between the two streams. Grubb and Dannenberg [GD97] find that successive note pairs having the same pitch are problematic for their statistical model of the problem. In order to emphasise the first of such successive sequences and harmonic change that is encountered, matches are penalised when there have been other similar notes recently in the stream. Thus we give priority to matches between notes which are novel occurrences in both streams. For a note of MIDI pitch n, occurring at

time T_n, $r(m,n)$ corresponds to the number of recent notes in that stream within the matching window w_{recent} that are the same or octaves of n:

$$r(m,n) = \begin{cases} 1 & \text{if } m = n \text{ and } |T_m - T_n| < w_{recent} \\ 0.5 & \text{if } |m-n| = 12 \text{ and } |T_m - T_n| < w_{recent} \\ 0 & \text{otherwise} \end{cases}.$$

Then we weight all occurrences over the window w_{recent}:

$$z(m,n) = \sum_m r(n,m) g(T_m, T_n, 0.5 w_{recent}), \tag{A.1}$$

and the 'surprise' function, $S(n)$, for the new note n is given by:

$$S(n) = 0.5^{z(m,n)} \tag{A.2}$$

The window w_{recent} determining what constitutes the recent past is dynamically changed so that the median is 0.7. The resulting match measure is:

$$M(n_l, n_a) = S(n_l).S(n_a).V(n_l).V(n_a).P(n_l, n_a) \tag{A.3}$$

where n_l and n_a are the MIDI pitches of the live and accompaniment streams respectively, and $P(n)$ measures whether the notes are identical or related as octaves:

$$P(n_l, n_a) = \begin{cases} 1 & \text{if } n_l = n_a \\ 0.5 & \text{if } |n_l - n_a| = 12 \\ 0.25 & \text{if } |n_l - n_a| = 24 \\ 0 & \text{otherwise} \end{cases}.$$

V(n) is the normalised volume which provides a bias towards louder notes. For all matches satisfying $M(n_l, n_a) > \theta_{match}$, where θ_{match} is a dynamic threshold, we calculate the alignment time, defined as the time difference between the two events:

$$A_n = T_{n_a} - T_{n_l}, \tag{A.4}$$

where T_{n_l} and T_{n_a} are the times of the live note and accompaniment note respectively.

The algorithm has a running estimate for the alignment which is used to select from the wide number of potential matches resulting from equation A.3. We calculate a non-normalised Gaussian of the observation with respect to our current estimate for the alignment, $\overline{A_n}$ (described below). The standard deviation which reflects uncertainty in the matching procedure and is a parameter that can be updated to change the algorithm's response. Thus, we define: $g(x, \mu, \sigma) = e^{-\frac{(x-\mu)^2}{2\sigma^2}}$.

Then, if $M(n_l, n_a).g[A_n, \overline{A_n}, \sigma_{align}] > \theta_{align}$, we accept the match and update our estimates for the alignment.

A.1.3 Tempo Adjustment

The tatum is defined by Bilmes [Bil93] as the smallest temporal atom. Here we adopt the term, although τ_{acc} can be assumed to refer to the duration of an eighth note of the accompaniment stream. For each alignment, A_k, we will make an adjustment to the tatum of $\Delta\tau_k$ milliseconds. Then the effect of these recent adjustments will have accumulated, so that for a previously observed alignment of time difference, A_k, our prediction for the current alignment is:

$$E[A_n|A_k] = A_k + \sum_{p=k}^{n-1} \Delta\tau_p . \frac{T_n - T_p}{\tau_{acc}} \tag{A.5}$$

We can now calculate the expected alignment, which we shall denote $\overline{E[A_n]}$, by summing the expectation over all recent alignments and weighting the expectations:

$$\overline{E[A_n]} = \frac{\sum_{k=1}^{K} M_k r_k g(A_n, E[A_n|A_k], \sigma_x) E[A_n|A_k]}{\sum_{k=1}^{K} M_k r_k g(A_n, E[A_n|A_k], \sigma_x)} \tag{A.6}$$

By including the measure term, M_k, we favour those matches which had higher measure according to equation A.3. The sum over the recent K terms is determined by a window within which we will use previous observations. We used the most recent 15 observations and weighted them according to how recent they are by an additional function r, defined by:

$$r(k) = g(t_n - 600, t_k, 4000). \tag{A.7}$$

The Gaussian, $g(A_n, E[A_n|A_k], \sigma_x)$ biases our mean expectation to those observations in the past which agree with our current observation. Methods used to adjust σ_{align} dynamically are described below.

From these expectations, we can generate a running estimate of the alignment difference, $\overline{A_n}$, using an exponential moving average:

$$\overline{A_n} = \alpha\overline{E[A_n]} + (1-\alpha)\overline{A_{n-1}}. \tag{A.8}$$

We wish to minimise $\overline{A_n}$ whilst also minimising change to the tempo and taking into account that our estimate may not be exact. Early attempts to synchronise purely on this estimated alignment difference exhibited a tendency to oscillate between positive and negative phase difference. Only by modelling the relative position *and* tempo of the two streams can suitable adjustments be made to bring about synchronisation. The aim of the decision process is to minimise such fluctuation whilst maintaining a responsiveness in the tracker so that alignment differences are rectified. We thus require an estimate for the change to the average alignment for recent matches since the relative tempo will have considerable effect on our synchronisation decision. Thus,

$$E[\Delta\overline{A_n}|\overline{A_k}] = \frac{\overline{A_n} - E[\overline{A_n}|\overline{A_k}]}{T_n - T_k}\tau_{acc} + \sum_{p=k}^{n-1}\Delta\tau_p, \tag{A.9}$$

where $E[\overline{A_n}|\overline{A_k}]$ is defined as in equation A.5, but substituting the alignment estimate, $\overline{A_k}$, for the observation A_k. The combination of estimation procedures gives rise to two estimates: the alignment difference, $\overline{A_n}$, and the estimated relative tempo, $\Delta\overline{A_n}$.

In order to adjust the alignment, we have chosen to make a suitable compensation so that the alignment prediction three bars in the future is zero according to our alignment and relative tempo estimate. This assumes that the change in relative tempo is zero and the three bar limit (twenty four tatum intervals) was chosen so that phase rectification is fast but not too sudden.

$$\Delta\tau_n = -\frac{(\overline{A_n} + 24\Delta\overline{A_n})}{24} \tag{A.10}$$

Then we set τ_{acc} to be $\tau_{acc} + \Delta\tau_n$. This tempo change is limited to a maximum of 6 BPM over three seconds in order to prevent any extreme

fluctuations. This can be adjusted if necessary, although since the pre-recorded audio is expected to be an alternative take, then we expect the two streams to be commensurate in tempo.

A.1.4 Adaptation of System Parameters

Having used Gaussians to model the expected prior probability distribution around our mean estimate, we require a rule to adapt this distribution dependent upon the data. The standard deviation reflects our uncertainty in the estimate, so where the observation supports the estimate, the standard deviation is decreased. When a new observation agrees with our estimate enough to be considered valid, but the alignment gaussian is low, then we increase the standard deviation.

A formal solution to this problem may require a more complete Bayesian formulation, but this could also be computationally problematic in a real-time system and require training. We describe how an estimate obtained from dynamic programming can be used below.

A.2 Evaluation

Evaluation is a difficult issue for real-time systems since we are interested in the subjective experience of performing with the system rather than statistical measurements obtained in tests. However, the use of recorded audio does provide an objective measure in the absence of interaction. We tested the algorithm by attempting to synchronise recordings of pieces from Bach's Well-Tempered Clavier performed by Friedrich Gulda to renditions by Keith Jarrett. Ideally, the accompaniment would consist of a previous rendition by the same performer so that the recorded performance contains similar stylistic deviations of tempo and expressive timing that the musician makes during the piece. The use of different performers is a stricter test of the algorithm. The system was set with an initial tempo gauged by ear so the two tracks were approximately synchronised and the MIDI value of the first note to trigger the accompaniment. Due

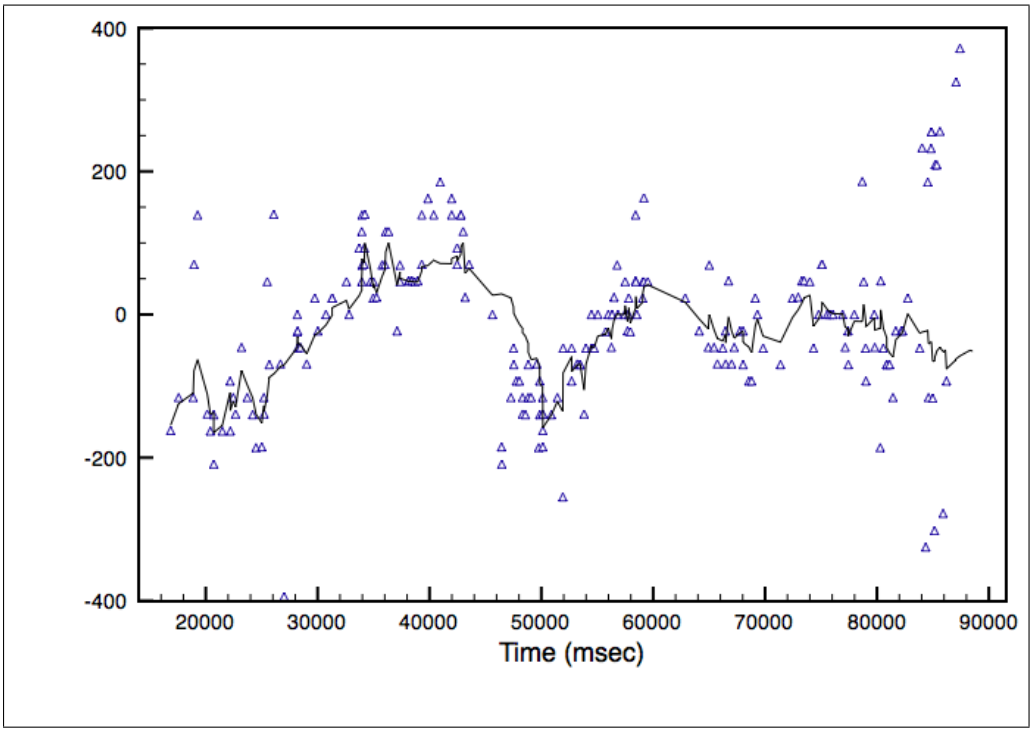

Figure A.3: Observations above the accuracy threshold in the alignment window (triangles) and the resulting alignment estimate (solid) of the algorithm for Prelude No. 9 of Bach's Well-Tempered Clavier.

to the minor latency involved in starting, we generally observed the accompaniment speed up to 'catch' the accompaniment audio and then slow to achieve synchronisation as can be seen in Figure A.3.

Goto and Muraoka [GM97] suggest the use of the Longest Continuous Segment for beat tracking evaluation. We indicate whether there was an adequate synchronisation was achieved. Out of the pieces tested, 21 of 24 were successfully followed from beginning to end. Subjectively the algorithm seems successful at synchronising the two audio files and listening to them simultaneously, one hears shifts in phase between the performances. In over half the pieces tested, the median and the mean were under 100ms. Given that there is no auditory feedback or genuine interaction, this is an impressive result.

The three cases where synchronisation failed demonstrated potential improvements that could be made to the algorithm. The common feature was a divergence in tempo between the two performers that was beyond

Results	Aligned	Mean	Median	Maximum
Prelude No.1	Y	55	40	240
Fugue No.1	Y	93	60	240
Prelude No.2	N	4129	1840	17000
Fugue No.2	Y	20	20	40
Prelude No.3	Y	49	60	140
Fugue No.3	Y	38	20	140
Prelude No.4	Y	60	135	460
Fugue No.4	Y	142	120	400
Prelude No.5	Y	64	60	300
Fugue No.5	Y	298	280	740
Prelude No.6	Y	83	40	460
Fugue No.6	Y	233	100	1500
Prelude No.7	N	2180	4027	18420
Fugue No.7	Y	54	60	120
Prelude No.8	N	1123	440	6320
Fugue No.8	Y	135	60	660
Prelude No. 9	Y	54	40	180
Fugue No.9	Y	29	20	80
Prelude No.10	Y	592	100	2760
Fugue No.10	Y	174	200	260
Prelude No.11	Y	70	40	260
Fugue No.11	Y	65	40	400
Prelude No.12	Y	235	200	640
Fugue No.12	Y	219	60	1060

Table A.1: Results showing the errors (in msec) from synchronising pieces from Bach's Well-Tempered Clavier by Friedrich Gulda to recordings by Keith Jarrett. Overall synchronisation is indicated by a Y or N in the second column.

the scope of the algorithm to compensate for. Whilst our assumptions stated that the piece should provide a suitable prior estimation for the future event distribution if played at identical tempos, by using renditions of the same piece by different musicians, the challenge to synchronise was thereby harder than in the case when our assumption holds true. However, divergence through tempo deviation is likely to occur in a performance and the system clearly requires a means of error recognition and recovery. This is discussed below.

We recorded the same statistics as used by Dixon [Dix05] for offline

evaluation of his MATCH algorithm: the mean, median and maximum error in alignment during each piece. These exclude a short (14 second) segment at the beginning and end of each piece to account for the problems of initial synchronisation and the tendency for dramatic ritardando at the end of pieces. If included, these have the potential to be misleading.

In order to generate statistical data, we performed an alignment in Sonic Visualiser [CLSB06] using Dixon's MATCH algorithm. This provides us with annotated 'ground-truth' data from which the mean, median and maximum were extracted. There could be some error introduced by this offline alignment process, but generally it is expected that this would be minimal compared to the real-time alignment differences experienced. The results are tabulated in Table A.1. The degree of synchronisation appears acceptable and agrees with our aural perception that the algorithm succeeds in real-time synchronisation and has strong potential as an interactive performance tool.

A.2.1 Error Recovery

There are several potential pitfalls for a system which seeks to classify observations and make explicit calculations in this manner. It would difficult to train such a system using 'ground-truth' data since the algorithm adjusts the accompaniment in real-time hence affecting the tempo. In addition, there may be no optimal tempo change in a given situation. Since our aim is for the accompaniment to be aesthetically acceptable, we have attempted a balance between fast synchronisation from perceived error and tempo smoothness. Potentially offline algorithms could test the least squared error that could result from a line of best fit and the algorithm could be evaluated with respect to this bound, and this is a potential area for inquiry.

The major problem faced by our algorithm is that there is no provision to check whether the current estimate is indeed approximately correct *across* the range of potential values. We rely on the assumption that our alignment estimate is in the general locality of the optimal alignment so that the matches used by the algorithm are for note events that ought

Pitch values	C#5	F#6	B6	D5	G5	**C#5**
C#6	29.74	30.01	30.28	30.01	30.28	30.01
D5	29.47	29.74	30.01	31.28	31.01	30.73
G5	29.20	29.47	29.74	31.01	32.28	32.01
C#5	29.93	29.66	29.47	30.74	32.01	**33.28**
G#6	29.66	29.93	29.66	30.47	31.74	33.01
B4	29.39	29.66	29.93	30.20	31.47	32.74
C#6	29.20	29.39	29.66	29.93	31.20	32.47

Table A.2: Dynamic Programming on two pitch sequences within our alignment evaluation. The winning alignment is shown in bold.

to be synchronous. When this estimate was approximately correct, the tempo adjustments made often synchronised the audio with mean and median under 100 to 200 msec as seen in Table A.1. However, there was no mechanism to know when the observations had become less reliable and seek a new estimate.

One option might be to look at the density of notes and the density and accuracy of matches. However, when the audio-to-MIDI conversion is not necessarily accurate, there may be a low density of matches even when the alignment is correct. The problem is how to analyse the match across the whole range of potential values. Whilst generating a probability distribution for the alignment estimate is not possible in real-time, it is possible to calculate the optimal alignment according to a classic dynamic programming algorithm. In his 1984 ICMC paper, whilst using dynamic programming to calculate the alignment, Dannenberg [Dan84] also describes the method as a potential means by which a computer can recover from error. We implemented a secondary algorithm based on Dannenberg's that penalises added or skipped notes, and rewards correct matches, defined so that the alignment rating, $d(i, j)$, between observed notes $n_l(i)$ and $n_a(j)$ of the live and accompaniment accompaniment streams can be calculated

iteratively:

$$d(i,j) = \max\left\{ \begin{array}{l} d(i-1,j) - w, \\ d(i,j-1) - w, \\ d(i-1,j-1) + \delta(n_l(i), n_a(j)) \end{array} \right\}, \tag{A.11}$$

where w is the cost of a non-match, $d(0,0) = 0$ and $\delta(a,b)$ equals 1 if and only if a equals b. Table A.2 illustrates the algorithm used on our test data with the penalty w set to 0.27.

This can be applied to the adaptation of system parameters as described in section A.1.4. If the resulting estimate is outside the expected window around our alignment estimate, $\overline{A}_n$, then we increase the standard deviation, σ_{align}, and if it is inside then we decrease it. Although this estimate can fluctuate between values, this method sets an equilibrium for the expectation window in which matches are accepted so that the standard deviation used is the median error between the dynamic programming algorithm and the tracksuit algorithm described in section A.1.1

Currently we are investigating the use of the dynamic programming estimate for the purposes of automatic recovery so that the algorithm is robust when experiencing unexpected timing deviations between the live audio and accompaniment. One problem encountered by this form of dynamic programming is that the order of notes in chords can be reversed resulting in a less than optimal path cost for a correct match. A clustering algorithm has been suggested by Dannenberg although, in this context, it is less critical than when the algorithm's result is used to align the sequences directly. Initial results look promising for this dual system.

A.3 Discussion

Figure A.4 shows the algorithm's output when tracking a song on the guitar. The errors within our matching window are scattered more widely and thus providing less precision as to tempo and song position. Whilst tests with piano were moderately successful, this may be accounted for by the fact that there are many note variations that provide many alignment points for the algorithm. When guitar is used as input, the part often

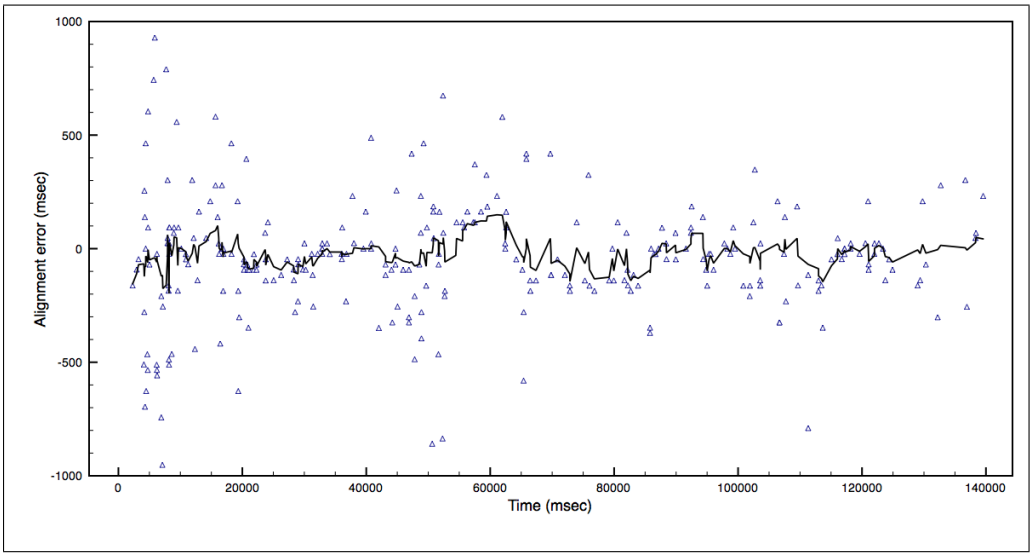

Figure A.4: Observations within the alignment window (triangles) and the resulting alignment estimate with a guitar part.

involves 'strumming' a sequence of chords.

The advantage in using MIDI representation is that it is computationally less demanding since the data is reduced to note-on and note-off timing information and volume. However, although it is successful at tracking piano which suits representation in this form, our audio-to-MIDI algorithm does not adequately represent the information in a guitar signal. Chromagram analysis [HS05] or chord recognition [SP09] might prove an alternative to MIDI representation for the guitar.

We aim to investigate the integration of several algorithmic techniques to create a more robust system. In particular, beat tracking could guide the tempo calculation process described above, whilst harmonic change could be used to determine the phase difference. Adaptation of Dixon's MATCH algorithm for real-time use would also provide an alignment estimate that could be used in this process either as an estimate or to adapt the behaviour of the system's parameters.

A.4 Summary

We have presented a synchronisation algorithm for audio sources which makes use of MIDI streams for the alignment process. The results indicate that symbolic MIDI may be usable for the purposes of synchronisation, provided these denote reliable salient onsets in the signal. We also aim to to contrast these results with those achieved by including other features such as spectral difference.

At present, our algorithm is not as reliable we would ideally hope for in a performance system. However, sequence alignment could be used to improve the algorithm, particularly in periods of uncertainty when it is searching for the best estimate. In these cases, sequence alignment could provide a coarse estimate of the probability distribution at different alignment times, so that closest to the current estimate could be chosen. By switching between two modes, the modelling algorithm could be used to adjust small variations in the tempo to optimise the synchronisation and the sequence alignment could search for a new estimate if matches no longer occur close to the estimate.

We aim to investigate similar approaches when using lower-level information. In using only high-level musical information, we inherently reduce the information content of the signal to a symbolic level, discarding some of the information present in the signal. However, if this symbolic level, focusing on pitched note onsets is the important feature that we require to be synchronised, then this has two advantages. We have less noise in our feature and the computation time is significantly improved since we calculate synchronisation on an event-based level rather than the frequency or time domain.

Bibliography

[AD90] P. E. Allen and R. B. Dannenberg. Tracking musical beats in real time. In *Proceedings of the 1990 International Computer Music Conference*, pages 140–143, 1990.

[Alé95] O. Alén. Rhythm as duration of sounds. *Tumba Francesca. Ethnomusicology*, 39(1):55–71, 1995.

[Ani97] S. Anisman. Lessons in listening. *Modern Drummer Magazine*, 1997.

[AP04] S. A. Abdallah and M. D. Plumbley. Polyphonic transcription by non-negative sparse coding of power spectra. In *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR 2004), Barcelona, Spain*, pages 318–325, 2004.

[Asc02] G. Ascherschleben. Temporal control of movements in sensorimotor synchronization. *Brain and Cognition*, 48(1):66–79, 2002.

[AvdP22] E. V. Appleton and B. van der Pol. On a type of oscillation-hysteresis in a simple triode generator. *Philosophy Magazine*, 43:177–193, 1922.

[BD86] W. Buxton and R. B. Dannenberg. The computer as accompanist. In *CHI '86: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 41–43, 1986.

[BDA+05] J. P. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, and M. Sandler. A tutorial on onset detection in music signals. *IEEE Transactions on Speech and Audio In Processing*, 13:1035–1047, 2005.

[Ben03] B. E. Benson. *The Improvisation of Musical Dialogue: A Phenomenology of Music*. Cambridge University Press, 2003.

[Ber94] P. F. Berliner. *Thinking In Jazz: The Infinite Art of Improvisation*. University of Chicago Press, 1994.

[Bil92] J. Bilmes. A model for musical rhythm. In *Proceedings of International Computer Music Conference*, pages 207–210, 1992.

[Bil93] J. Bilmes. Timing is of the essence: Perceptual and computational techniques for representing, learning, and reproducing expressive timing in percussive rhythm. Master's thesis, Massachusetts Institute of Technology, Cambridge, MA, 1993.

[Bil94] J. A. Biles. GenJam: A genetic algorithm for generating jazz solos. In *Proceedings of the International Computer Music Conference*, pages 131–137, 1994.

[Bil07] J. A. Biles. *Improvising with Genetic Algorithms: GenJam*, chapter 7. Springer, 2007.

[Bro93] J. Brown. Determination of the meter of musical scores by autocorrelation. *Journal of the Acoustical Society of America*, 94(4):1953–1957, 1993.

[BY04] T. Blackwell and M. Young. Swarm Granulator. *EvoWorkshops.* Springer-Verlag, 2004.

[CDW07] A. Cont, S. Dubnov, and D. Wessel. Realtime multiple-pitch and multiple-instrument recognition for music signals using sparse non-negative constraints. In *Proceedings of Digital Audio Effects Conference (DAFx)*, pages 85–92. Bordeaux, October 2007.

[Cem04] A. T. Cemgil. *Bayesian Music Transcription*. PhD thesis, Radboud Universiteit Nijmegen, 2004.

[CGLT04] C. Chafe, M. Gurevich, G. Leslie, and S. Tyan. Effect of time delay on ensemble accuracy. In *Proceedings of the International Symposium on Musical Acoustics, March 31st to April 3rd 2004 (ISMA2004), Nara, Japan*, 2004.

[CK03] A. T. Cemgil and B. Kappen. Monte Carlo methods for tempo tracking and rhythm quantization. *Journal of Artificial Intelligence Research*, 18:45–81, 2003.

[CLSB06] C. Cannam, C. Landone, M. B. Sandler, and J. Bello. The Sonic Visualiser: A visualisation platform for semantic descriptors from musical signals. In *Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR-06)*, 2006.

[CM63] G. Cooper and L. B. Meyer. *The Rhythmic Structure of Music.* University of Chicago Press, 1963.

[CMRW03] N. Collins, A. McLean, J. Rohrhuber, and A. Ward. Live coding in laptop performance. *Organised Sound*, 8(3):321–330, dec 2003.

[Coh05] P. R. Cohen. If not Turing's test, then what? *A.I. Magazine*, 26(4):61–67, 2005.

[Col05a] N. Collins. An automated event analysis system with compositional applications. In *Proceedings of the International Computer Music Conference, Barcelona*, 2005.

[Col05b] N. Collins. DrumTrack: Beat induction from an acoustic drum kit with synchronised scheduling. In *Proceedings of International Computer Music Conference*, 2005.

[Col06] N. Collins. *Towards Autonomous Agents for Live Computer Music: Realtime Machine Listening and Interactive Music Systems.* PhD thesis, Centre for Science and Music, Faculty of Music, University of Cambridge., 2006.

[Con06] A. Cont. Realtime multiple pitch observation using sparse non-negative constraints. In *International Symposium on Music Information Retrieval (ISMIR)*, pages 206–212. Victoria, CA., October 2006.

[Con08a] A. Cont. Antescofo: Anticipatory synchronization and control of interactive parameters in computer music. In *Proceedings of International Computer Music Conference (ICMC)*. Belfast, August 2008.

[Con08b] A. Cont. *Modeling Musical Anticipation: From the time of music to the music of time*. PhD thesis, University of Paris 6 and University of California in San Diego, October 2008.

[Cop96] D. Cope. *Experiments in Musical Intelligence*. Madison, WI: A-R Editions, 1996.

[Cop01] D. Cope. *Virtual Music: Computer Synthesis of Musical Style*. MIT, Cambridge, MA., 2001.

[CZS+05] E. Chew, R. Zimmermann, A. Sawchuk, C. Papadopoulos, C. Kyriakakis, C. Tanoue, D. Desai, M. Pawar, R. Sinha, and W. Meyer. A second report on the user experiments in the distributed immersive performance project. *5th Open Workshop of MUSICNETWORK: Integration of Music in Multimedia Applications*, 2005.

[Dah05] S. Dahl. *On the beat: Human movement and timing in the Production and Perception of Music*. PhD thesis, KTH - Royal Institute of Technology, Stockholm, 2005.

[Dan84] R. B. Dannenberg. An on-line algorithm for real-time accompaniment. In *Proceedings of the International Computer Music Conference, San Francisco*, pages 193–198, 1984.

[Dan97] M. Danks. Real-Time Image and Video In Processing in Gem. In *Proceedings of International Computer Music Conference*, pages 220–223, 1997.

[Dan05] R. B. Dannenberg. Toward automated holsitic beat tracking, music analysis and understanding. In *International Conference on Music Information Retrieval*, 2005.

[dCK02] A. de Cheveigné and H. Kawahara. Yin, a fundamental frequency estimator for speech and music. *The Journal of the Acoustical Society of America*, 111(4):1917–1930, 2002.

[DH99] P. Desain and H. Honing. Computational models of beat induction: A rule based approach. *Journal of New Musical Research*, 28(1):29–42, 1999.

[Dix00] S. Dixon. On the computer recognition of solo piano music. In *in Proceedings of Australasian Computer Music Conference*, pages 31–37, 2000.

[Dix01] S. Dixon. Automatic extraction of tempo and beat from expressive performances. *Journal of New Musical Research*, 30:39–58, 2001.

[Dix05] S. Dixon. Match: A music alignment tool chest. In *in Proc. ISMIR*, pages 492–497, 2005.

[DJB00] C. Drake, M. R. Jones, and C. Baruch. The development of rhythmic attending in auditory sequences: Attunement, referent period, focal attending. *Cognition*, 77(3):251–288, 2000.

[dlCMS01] P. de la Cuadra, A. Master, and C. Sapp. Efficient pitch detection techniques for interactive music. In *Proceedings of International Computer Music Conference*, 2001.

[DMR87] R. B. Dannenberg and B. Mont-Reynaud. Following an improvisation in real time. In *In Proc. of the 1987 International Computer Music Conference*, pages 241–248, 1987.

[Dob04] C. Dobrian. Strategies for continuous pitch and amplitude tracking in real-time interactive improvisation software. In

Proceedings of the 2004 Sound and Music Computing Conference (SMC04), IRCAM, Paris, France, 2004.

[Dow08] J. S. Downie. The music information retrieval evaluation exchange (2005–2007): A window into music information retrieval research. *Acoustical Science and Technology*, 29(4):247–255, 2008.

[DP05] M. Davies and M. D. Plumbley. Beat tracking with a two state model. In *Proc. IEEE International Conference on Acoustics, Speech and Signal In Processing (ICASSP), Philadeliphia, USA*, volume 3, pages 241–244, 2005.

[DP07] M. E. P. Davies and M. D. Plumbley. Context-dependent beat tracking of musical audio. *IEEE Transactions on Audio, Speech and Language Processing*, 15(3):1009–1020, 2007.

[Dun10] K. Dunlap. Reactions on rhythmic stimuli, with attempt to synchronize. *Psychological Review*, 17:399–416, 1910.

[DW96] V. D and A. Wing. Modeling variability and dependence in timing. In H. Heuer and S. Keele, editors, *Handbook of perception and action*, volume 2, pages 181–262. London: Academic Press, 1996.

[Ell07] D. P. W. Ellis. Beat tracking by dynamic programming. *Journal of New Musical Research*, 36(1):51–60, 2007.

[FL02] P. Freeman and L. Lacey. Swing and groove: Contextual rhythmic nuance in live performance. In *7th International Conference on Music Perception and Cognition*, 2002.

[For73] Forney. The Viterbi algorithm. *Proceedings of the IEEE*, pages 268–278, 1973.

[Fri99] A. Friberg. Jazz drummers' swing ratio in relation to tempo. Paper presented at the Acoustical Society of America ASA/EAA/DAGA meeting, 1999.

[GD97] L. Grubb and R. B. Dannenberg. A stochastic method of tracking a performer. In *Proceedings of the International Computer Music Conference*, pages 301–308, 1997.

[GD05] F. Gouyon and S. Dixon. A review of automatic rhythm description systems. *Computer Music Journal*, 29(1):34–54, 2005.

[GFB03] F. Gouyon, L. Fabig, and J. Bonada. Rhythmic expressiveness transformations of audio recordings: swing modifications. In *in Proceedings of the 6th Int. Conference on Digital Audio Effects (DAFX-03), London, UK*, pages 94–99, 2003.

[GM94] M. Goto and Y. Muraoka. A beat tracking system for acoustic signals of music. In *Proceedings of the Second ACM International Conference on Multimedia*, pages 365–372, 1994.

[GM97] M. Goto and Y. Muraoka. Issues in evaluating beat tracking systems. In *IJCAI-97 Workshop on Issues in AI and Music-Evaluation and Assessment Issues in Evaluating Beat Tracking Systems*, pages 9–16, 1997.

[GM99] P. D. Gader and M. Mystkowski. Applications of hidden Markov models to detecting landmines with ground penetrating radar. In *Proceedings of the SPIE Conferfence on Detection and Remediation Technologies for Mines and Minelike Targets IV Orlando, FL*, 1999.

[Got01] M. Goto. An audio-based real-time beat tracking system for music with or without drum sounds. *Journal of New Musical Research*, 30:159–171, 2001.

[Gra96] T. Gracyk. *Rhythm and Noise*. I. B. Tauris and Co. Ltd, 1996.

[Hai03] S. Hainsworth. Beat tracking with particle filtering algorithms. In *In IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 91–94, 2003.

[Hai06] S. Hainsworth. Signal processing methods for music transcription. In A. Klapuri and M. Davy, editors, *Beat Tracking and Musical Metre Analysis*, pages 101–129. Berlin: Springer, 2006.

[Han89] S. Handel. *Listening: An Introduction to the Perception of Auditory Events.* Bradford Books, 1989.

[Har00] S. Harnard. Minds, machines and Turing. *Journal of Logic, Language and Information*, 9(4):425–445, 2000.

[HBHK04] R. Hiraga, R. Bresin, K. Hirata, and H. Katayose. Rencon 2004: Turing test for musical expression. In *Proceedings of the 2004 international conference on New Interfaces for Musical Expression*, 2004.

[HH06] S. Hirsch and S. Heithecker. *Pro Tools 7 Session Secrets.* John Wiley and Sons, 2006.

[HS05] C. A. Harte and M. B. Sandler. Automatic chord identification using a quantised chromagram. In *Proceedings of the 118th Convention of the Audio Engineering Society, Barcelona, Spain*, 2005.

[Hur06] D. Huron. *Sweet Anticipation.* MIT Press, 2006.

[Huy86] C. Huygens. *Horologium oscillatorium.* republished in English by Iowa State College, Ames, 1673; reprinted 1986.

[Igo06] T. Igoe. In the Pocket. Essential Grooves. Part 2. Funk. *Modern Drummer*, July 2006.

[Iye98] V. Iyer. *Microstructures of Feel, Macrostructures of Sound: Embodied Cognition in West African and African-American Musics.* PhD thesis, University of California, Berkeley, 1998.

[Izh08] R. Izhaki. *Mixing Audio.* Focal Press, 2008.

[JB89] M. R. Jones and M. Boltz. Dynamic attending and responses to time. *Psychological Review*, 96(3):459–491, 1989.

[Jeh05] T. Jehan. Downbeat prediction by listening and learning. In *Proceedings of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA) Mohonk, NY*, 2005.

[Jon76] M. R. Jones. Time, our lost dimension: toward a new theory of perception, attention, and memory. *Psychological Review*, 83:323–355, 1976.

[Jor05] S. Jordà. *Digital Lutherie: Crafting musical computers for new musics' performance and improvisation*. PhD thesis, Universitat Pompeu Fabra, 2005.

[JS01] T. Jehan and B. Schoner. An audio-driven perceptually meaningful timbre synthesizer. In *Proceedings of International Computer Music Conference*, pages 381–388, 2001.

[Jus02] T. Justus. Musical Jabberwocky? *Trends in Cognitive Sciences*, 6(3):144–145, 2002.

[KEA06] A. P. Klapuri, A. J. Eronen, and J. T. Astola. Analysis of the meter of acoustic musical signals. In *IEEE Transactions on Speech and Audio Processing*, pages 342–355, 2006.

[Kla03] A. P. Klapuri. Multiple fundamental frequency estimation by harmonicity and spectral smoothness. *IEEE Transactions on Speech and Audio Processing*, 11(6):804–816, 2003.

[Kla06] A. P. Klapuri. Multiple fundamental frequency estimation by summing harmonic amplitudes. In *7th International Conference on Music Information Retrieval (ISMIR-06), Victoria, Canada*, 2006.

[Kro99] M. Krol. Have we witnessed a real-life Turing Test? *Computer*, 32(3):27–30, Mar 1999.

[Lai00] J. E. Laird. Creating human-like synthetic characters with multiple skill levels: A case study using the soar quakebot. In

in Proceedings of the AAAI Fall Symposium Technical Report, pages 54–58, 2000.

[Lar95] E. W. Large. Beat Tracking with a Nonlinear Oscillator. In *Working Notes of the IJCAI-95 Workshop on Artificial Intelligence and Music, Montreal*, 1995.

[Lar03] J. Laroche. Efficient tempo and beat tracking in audio recordings. *Journal of the Audio Engineering Society*, 51(4):226–233, 2003.

[Lei73] S. Leigh. *Paul Simon. Now and Then.* Raven Books, 1973.

[Lev97] D. J. Levitin. Still creative after all these years: A conversation with Paul Simon. *Grammy Magazine*, 1997.

[LH07] D. Litke and K. Hamel. A score-based interface for interactive computer music. In *Proceedings of the International Computer Music Conference*, 2007.

[Liv06] D. Livingstone. Turing's Test and believable AI in games. *ACM Computers In Entertainment*, 4(1), 2006.

[LJ83] F. Lerdahl and R. Jackendoff. *A Generative Theory of Tonal Music.* MIT, Cambridge, MA., 1983.

[LJ99] E. W. Large and M. R. Jones. The dynamics of attending: How people track time-varying events. *Psychological Review*, 106(1):119–159, 1999.

[LK94] E. W. Large and J. F. Kolen. Resonance and the perception of musical meter. *Connection Science*, 6(2):177–208, 1994.

[LK04] N. P. Lago and F. Kon. The quest for low latency. In *Proceedings of the International Computer Music Conference*, pages 33–36, 2004.

[Mar05] M. Marolt. A connectionist model of finding partial groups in music recordings with application to music transcription. In R. et al., editor, *Adaptive and natural computing algorithms :*

Proceedings of the International Conference in Coimbra, Portugal, pages 494–497, 2005.

[Mat94] J. Mates. A model of synchronization of motor acts to a stimulus sequence I. Timing and error corrections. *Biological Cybernetics*, 70(5):463–473, 1994.

[McA95] J. D. McAuley. *Perception of Time as Phase: Toward an Adaptive Oscillator Model of Rhythmic Pattern Processing.* PhD thesis, Indiana University, 1995.

[McC96] J. McCartney. SuperCollider: A new real-time synthesis language. In *Proceedings of the International Computer Music Conference*, pages 257–258, 1996.

[McG06] A. McGuiness. Groove microtiming deviations as phase shifts. In *9th International Conference on Music Perception and Cognition*, 2006.

[Mey56] L. B. Meyer. *Emotion and Meaning in Music.* Chicago: Chicago University Press, 1956.

[MMDK07] M. F. McKinney, D. Moelants, M. E. P. Davies, and A. Klapuri. Evaluation of audio beat tracking and music tempo extraction algorithms. *Journal of New Musical Research*, 36(1):1–16, 2007.

[MPH04] T. Mäki-Patola and P. Hämäläinen. Latency tolerance for gesture controlled continuous sound instrument without tactile feedback. In *Proceedings of the International Computer Music Conference (ICMC 2004), Miami, USA*, pages 409–416, 2004.

[OF01] N. Orio and FrançoisDechelle. Score following using spectral analysis and hidden Markov models. In *Proceedings of the International Computer Music Conference, Havana*, pages 105–109, 2001.

[OLS03] N. Orio, S. Lemouton, and D. Schwarz. Score Following: State of the art and new developments. In *Proceedings of*

the 2003 Conference on New Interfaces for Musical Expression (NIME), Montreal, Canada, pages 36–41, 2003.

[Pac02] F. Pachet. Interacting with a musical learning system: The Continuator. In *In Proceedings of the Second International Conference on Music and Artificial Intelligence, Edinburgh, Scotland (ICMAI 2002)*, pages 119–132, 2002.

[PAZ98] M. Puckette, T. Apel, and D. Zicarelli. Real-time audio analysis tools for Pd and MSP. In *Proceedings of International Computer Music Conference, San Francisco.*, pages 109–112, 1998.

[PE07] G. E. Poliner and D. P. W. Ellis. A discriminative model for polyphonic piano transcription. *EURASIP Journal on Applied Signal Processing*, 2007, 2007.

[PI08] A. Pertusa and J. M. Inesta. Multiple fundamental frequency estimation using gaussian smoothness. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing*, pages 105–108, 2008.

[PJR97] J. Pressing and G. Jolley-Rogers. Spectral properties of human cognition and skill. *Biological Cybernetics*, 76(339-347), 1997.

[Pop59] K. Popper. *The Logic of Scientific Discovery*. Basic Books, New York, NY, 1959.

[Pre02] J. Pressing. Black Atlantic rhythm: Its computational and transcultural foundations. *Music Perception*, 19(3):285–310, 2002.

[Puc96] M. Puckette. Pure Data: another integrated computer music environment. In *Proc. the Second Intercollege Computer Music Concerts, Tachikawa. Reprinted as* `ftp://crca-ftp.uscd.edu:~/pub/msp/pd-kcm.ps`, pages 37–41, 1996.

[Puc02] M. Puckette. Max at seventeen. *Computer Music Journal*, 26(4):31–43, 2002.

[PW01] M. Pearce and G. Wiggins. Towards a framework for the evaluation of machine compositions. In *In Proceedings of the AISB'01 Symposium on Artificial Intelligence and Creativity in the Arts and Sciences*, pages 22–32, 2001.

[Rab89] L. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[Rap99] C. Raphael. Automatic segmentation of acoustic musical signals using hidden Markov models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(4):360–370, apr 1999.

[Rap01] C. Raphael. Synthesizing musical accompaniments with Bayesian Belief Networks. *Journal of New Musical Research*, 30(1):59–70, 2001.

[Rap02] C. Raphael. A Bayesian Network for real-time musical accompaniment. In T. Dietterich, S. Becker, , and Z. Ghahramani, editors, *Advances in Neural Information In Processing Systems, NIPS 14*. MIT Press, 2002.

[Rap04] C. Raphael. Musical accompaniment systems. *Chance Magazine*, 17(4), 2004.

[Rep00] B. H. Repp. Compensation for subliminal timing perturbations in perceptual-motor synchronization. *Psychological Research*, 63(2):106–128, 2000.

[Rep01] B. H. Repp. Phase correction, phase resetting, and phase shifts after subliminal timing perturbations in sensorimotor synchronization. *Journal of Experimental Psychology: Human Perception and Performance*, 27(3):600–621, 2001.

[Rep03] B. H. Repp. Rate limits in sensorimotor synchronization with auditory and visual sequences: The synchronization threshold and the benefits and costs of interval subdivision. In *Journal of Motor Behavior*, pages 355–370, 2003.

[Rep05] B. H. Repp. Sensorimotor synchronization: a review of the tapping literature. *Psychonomic Bulletin and Review*, 12(6):969–992, 2005.

[RGD+93] R. Rowe, B. Garton, P. Desain, H. Honing, R. Dannenberg, D. Jacobs, S. T. Pope, M. Puckette, C. Lippe, Z. Settel, and G. Lewis. Editor's notes: Putting Max in perspective. *Computer Music Journal*, 17(2):3–11, 1993.

[Ros92] D. Rosenthal. Emulation of human rhythm perception. *Computer Music Journal*, 19(1):64–76, 1992.

[Row93] R. Rowe. *Interactive Music Systems*. The MIT Press, 1993.

[Row01] R. Rowe. *Machine Musicianship*. The MIT Press, 2001.

[SB03] P. Smaragdis and J. C. Brown. Non-negative matrix factorization for polyphonic music transcription. In *In IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 177–180, 2003.

[SBS+05] N. Schnell, R. Borghesi, D. Schwarz, F. Bevilacqua, and R. Muller. FTM - complex data structures in Max. In *Proceedings of the International Computer Music Conference*, 2005.

[Sch98] E. D. Scheirer. Tempo and beat analysis of acoustic musical signals. *The Journal of the Acoustical Society of America*, 103(1):588–60, 1998.

[SDP08] A. M. Stark, M. E. P. Davies, and M. D. Plumbley. Rhythmic analysis for real-time audio effects. In *Proceedings of the International Computer Music Conference (ICMC 2008), Belfast, Northern Ireland*, 2008.

[Sea80] J. Searle. Minds, Brains and Programs. *Behavioural and Brain Sciences*, 3:417–457, 1980.

[Shi94] S. M. Shieber. Lessons from a restricted Turing test. *Communications of the Association for Computing Machinery*, 37(6):70–78, 1994.

[SP09] A. M. Stark and M. D. Plumbley. A real-time frame level chord recognition algorithm. In *Proceedings of International Computer Music Conference, Montreal, Canada*, 2009.

[SS00] P. Sprent and N. C. Smeeton. *Applied Nonparametric Statistical Method*. CRC Press Inc., 3rd edition, 2000.

[SVS00] A. Semjen, D. Vorberg, and H. Schulze. Timing precision in continuation and synchronization tapping. *Psychological research*, 63(2):137–47, 2000.

[Tem99a] D. Temperley. Modeling meter and harmony: A preference-rule approach. *Computer Music Journal*, 23(1):10–27, 1999.

[Tem99b] D. Temperley. Syncopation in rock: A perceptual perspective. *Popular Music*, 18(1):19–40, 1999.

[TLO02] N. P. M. Todd, C. S. Lee, and D. J. O'Boyle. A sensorimotor theory of temporal tracking and beat induction. *Psychological Research*, 66(26-39), 2002.

[Toi98] P. Toiviainen. An interactive MIDI accompanist. *Computer Music Journal*, 22(4):63–75, 1998.

[TS03] P. Toiviainen and J. S. Snyder. Tapping to bach: Resonance-based modeling of pulse. *Music Perception*, 21(1):43–80, 2003.

[Tur50] A. Turing. Computing Machinery and Intelligence. *Mind*, 59:433–460, 1950.

[VE90] B. Vercoe and D. Ellis. Real-time Csound: Software synthesis with sensing and control. In *Proceedings of the 1990 International Computer Music Conference*, pages 209–211, 1990.

[Ver84] B. Vercoe. The Synthetic Performer in the context of live performance. In *Proceedings of the International Computer Music Conference (ICMC)*, pages 199–200, 1984.

[Vit67] A. J. Viterbi. Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. *IEEE Transactions on Information Theory*, pages 260–269, 1967.

[VP85] B. Vercoe and M. Puckette. Synthetic Rehearsal, training the Synthetic Performer. In *Proceedings of the International Computer Music Conference (ICMC 1985)*, pages 275–278, 1985.

[Waa01] C. H. Waadeland. "It Don't Mean a Thing If It Ain't Got That Swing" - Simulating expressive timing by modulated movements. *Journal of New Music Research*, 30(1):23–37, 2001.

[WC03] G. Wang and P. M. Cook. ChucK: A programming language for on-the-fly, real-time audio synthesis and multimedia. In *Proceedings of the International Computer Music Conference, Singapore.*, 2003.

[WCF+99] M. Wright, A. Chaudhary, A. Freed, S. Khoury, and D. Wessel. Audio applications of the sound description interchange format standard. In *Audio Engineering Society 107th Convention*, 1999.

[Wic90] P. Wicke. *Rock Music: Culture, Aesthetics and Sociology.* Cambridge University Press, 1990.

[WK73] A. M. Wing and A. B. Kristofferson. Response delays and the timing of discrete motor responses. *Perception and Psychophysics*, 14(1):5–12, 1973.

[WO02] M. M. Wanderley and N. Orio. Evaluation of input devices for musical expression: Borrowing tools from HCI. *Computer Music Journal*, 26(3):62–76, 2002.

[WS05] X. Wen and M. B. Sandler. A partial searching algorithm and its application for polyphonic music transcription. In *In Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR 2005)*, pages 690–695, 2005.

[Zic98] D. Zicarelli. An extensible real-time signal processing environment for Max. In *Proceedings of the International Computer Music Conference, Ann Arbor, MI.*, pages 463–466, 1998.

[ZR08] R. Zhou and J. D. Reiss. A real-time polyphonic music transcription system. In *Proceedings of the Fourth Music Information Retrieval Evaluation eXchange (MIREX 2008), Philadelphia, USA*, 2008.