Real-time Beat Prediction in Digital Music

by

Robert Douglas Harper

A thesis
presented to the University of Waterloo
in the fulfilment of the
thesis requirement for the degree of
Master of Applied Science
in
Systems Design Engineering

Waterloo, Ontario, Canada, 2004

I hereby declare that I am the sole author of this thesis.

I authorize the University of Waterloo to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize the University of Waterloo to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

The University of Waterloo requires the signatures of all persons using or photocopying this thesis. Please sign below, and give address and date.

Abstract

Music is a very complex audio signal essentially containing a cacophony of different sounds, yet, incredibly the human brain is able to process this signal and extract information such as melody, harmony, tone, expressiveness, and rhythm. One facet that is common to most music listeners, regardless of musical training, is the inherent ability to feel the beat of the music and express this feeling through a tapping foot or a dancing body. While this ability comes naturally to humans, it poses a significant challenge to mimic this ability using computational devices.

A number of algorithms, both real-time and offline, have been developed to detect the beat in a musical audio signal. Previous approaches use a wide variety of techniques including self-similarity, probabilistic measures, and connectionist models, with varying degrees of success. The purpose of this thesis is to develop and explore a new real-time capable connectionist model for automatic beat detection. This new approach is centred on competing, connected, self-adjusting and self-evaluating agents that attempt to find the correct beat period and locations. Each agent maintains its own possible continually-updated beat hypothesis that is evaluated using a precisely tuned enhanced recurrent timing network.

The proposed model is evaluated and compared to a leading beat detection approach on wide variety of musical genres using an assortment of quantitative and qualitative performance measures. The performance of the proposed system was found to be reasonably good, correctly predicting 70% of all beats in the corpus while only 12% of predicted beats were incorrect. Two persistent problems plagued the system resulting in slightly lower performance than was measured in the competing model. These problems were the tendency for the system to oscillate between two valid metrical levels of beat prediction output and the system's propensity for detecting only the off-beat. Nonetheless, the performance of the system is promising and warrants further research effort.

Acknowledgements

There are number of people who deserve a lot of credit for aiding and abetting me in the completion of this thesis. First and foremost, the academic and intellectual support provided by my supervisor, Dr. Ed Jernigan, was unparalleled and warrants great thanks. I also wish to thank my readers, Dr. Paul Fieguth and Dr. George Freeman, for their helpful suggestions, comments, critiques and criticism. Rachelle Eisen and my parents, Jan and Doug Harper, deserve unending gratitude for their constant moral support and the occasional kick in the right direction. I must give credit to Vicky Lawrence for all of her hard work in keeping my degree moving forward. Thanks must also go to many of my friends who pretended to care enough to listen to an explanation of the work contained herein. Finally, without John Medeski, Billy Martin, Chris Wood, Eric Krasno, Alan and Neal Evans, John Scofield, Oscar Peterson, Herbie Hancock, Grant Green, Charles Mingus, Ernest Ranglin, and Box-Car Willie, this thesis never would have been put down on paper.

Table of Contents

Table of Figures

Chapter 1: Introduction

"Music is the pleasure the human mind experiences from counting without being aware that it is counting." - Gottfried Leibniz (1646-1716)

Music has been an integral part of nearly every culture for thousands of years. As a means of communication, a form of entertainment or an art form, music seems to be a part of the human spirit. The tapping of a foot, the clapping of hands, and the steps of a dance all demonstrate the need for the listener to synchronize his/her movements with the sound he/she is hearing. Leibniz was astute in identifying this effortless synchronization of movement as a form of subconscious counting. This mental process finds humans following the pulse, the rhythm, the beat of the music.

The innate, natural ability that humans have for following the beat is quite remarkable. Music, as a signal, is very complex and literally contains a cacophony of sounds occurring at different times and with different pitches and textures. Incredibly, humans are able to listen to this information-rich signal and extract melody, feeling, rhythm, all the while tapping with the beat; we can hear structure in the complexity. The structured beat in the music can be more than felt or heard, it can be anticipated or predicted such that movements can be synchronized with beats not yet heard. Remarkably, this talent is found in most people, regardless of musical training.

While humans find it natural and often effortless to detect the beat in a musical stimulus, reproducing this behaviour using computational devices poses a significant challenge. Beyond the complexity of the signal itself, music can be very inexact, with continual tempo changes, timing inaccuracies, and notes played significantly off the beat. Many approaches to the problem of beat detection and prediction have been taken with varying degrees of success but an ideal solution has yet to be found. So far, a model that is able to match the performance of a human subject exists purely in theory.

It is the goal of this work to find a new and promising method for beat prediction that comes closer to the ultimate goal of perfect, human-comparable beat detection.

Many existing beat detection systems that find high degrees of success only process vastly simplified, symbolic forms of music, such as MIDI. The method proposed here attempts to not only detect the beat, but also predict future locations of the beat. Moreover, this new model also joins the ranks of systems that process the complex waveforms of acoustic musical signals. The proposed system provides a success-oriented new direction and insight into some of the challenging problems that face beat prediction systems.

1.1 Applications

There are a large number of potential and existing application areas for a successful beat detection or prediction system. These areas include cognitive modelling, performance analysis, automatic music processing and classification, and music to event synchronization.

Cognitive science circles often approach the problem from the standpoint of modelling human behaviour and thought processes (see [1-3]). These models can be used to further the understanding of the way in which the human mind works and how humans may *feel* the beat in music. This is not the approach taken here and thus the application to cognitive modelling is not considered further.

Another academic application of a successful beat detection or prediction algorithm is recorded performance analysis and ethno-musicological studies. The tempo and the location of the beat within the piece under consideration can be used as a basis for analysis and comparison of rhythmic content between performances, styles and cultures.

An industrial use for tempo and beat location information is the formation of the front end of an automatic music-processing system. Beat location could be used in algorithms such as time expansion and compression (slowing or hastening the tempo of a piece of music without affecting its pitch), enabling two pieces of music to be

matched in tempo. The same beat and tempo information could also be used as features for a music classification and retrieval database system.

The synchronization of events to music is a vast application area for beat detection and prediction systems. Detection of the location of the beat within a song could enable the synchronization of video events to the accompanying music. Beat-aligned scene changes and animated characters that move in time to the beat are just two examples of video event synchronization. Real-time applications, such as flashing lights coincident with the beat at a dance club, can also make use of beat prediction systems. One of the most interesting real-time applications of beat prediction is the automatic generation of accompaniment that follows a live musician. As Scheirer [4] states, musicians find it much easier to set the tempo themselves than to follow the tempo of someone else. A real-time beat prediction system could predict the locations of the beat given the musician's performance and provide timing direction to an auto-accompaniment generator, all the while allowing the human musician to set the tempo.

Each application area has a different set of characteristics that it dictates are most important and therefore differs in its measure of algorithm success. The model described in this paper focuses on providing a robust model for real-time event to music synchronization applications. For this reason, the proposed system is designed to perform beat prediction, be real-time capable, and take acoustic audio data as input.

1.2 Thesis Layout

The remainder of this document is structured as follows:

- Chapter 2: Theoretical Background. A brief description of rhythm and meter is provided in order to better define the problem of beat prediction. A more in depth specification of the proposed research follows.

- Chapter 3: Previous Models. Approaches to beat detection taken by other authors are discussed in relation to the system described here.

- Chapter 4: Recurrent Timing Networks. The basic element used for beat detection in the proposed system is discussed in detail in this chapter.

- Chapter 5: The Proposed Model. The design and implementation of the entire proposed system is described in detail.

- Chapter 6: Results and Analysis. Results generated by the proposed model and a competing model are given with corresponding analyses.

- Chapter 7: Recommendations and Conclusions. Concluding remarks regarding the performance of the model, success of the research, and future directions are given here.

Chapter 2: Theoretical Background

When designing an algorithm for beat detection or prediction, it is important to first specifically define what is meant by the term "beat". In the previous section, the beat in a piece of music was informally defined as the location that a listener may choose to tap his foot or clap her hands. While this makes logical sense, a more precise definition is required before moving forward. This chapter will cover a brief theoretical background to rhythmic structure and beat, and then proceed to clearly outline the goals of the proposed model within this theoretical framework.

2.1 Rhythmic Structure and Metrical Hierarchy

When listening to music, one is bombarded by a slew of auditory stimuli. Collections of the perceived sounds are subconsciously grouped together into sets such as motives, themes, and phrases that constitute the song. Many events within this stream of sounds are emphasized and draw attention to their point in the musical flow. These events are given the term *phenomenal accents* by Lerdahl and Jackendoff in their groundbreaking book "A Generative Theory of Tonal Music" [5]. A phenomenal accent can occur in many forms including emphasis through attack points or onsets of sounds or notes, sudden changes in sound, jumps in relative pitch, and longer notes [5].

Using the cues of these phenomenal accents, the listener attempts to create a best-fit mental metrical grid upon which perceived notes are placed [5]. This so-called metrical grid is composed of regularly spaced points in time, or beats. These beats have location but do not have duration and are conceptually similar to an impulse. In this way, beats mark points in time at which there exists a metrical accent – an emphasized instant due to temporal location in the metrical structure and not necessarily corresponding to a particular auditory event. Once this mental grid is constructed, it is strengthened by coincident phenomenal and metrical accents and only discarded in the face of strong contradictory evidence.

Metrical accents do not all have the same stress – a fact to which most music listeners will attest – but contain a strong-weak alternating pattern. For example, in a song with time signature *4/4* (four quarter note beats in each measure), the first and third beats in the bar have a stronger weight than the second and fourth. Moreover, the first beat is in-turn stronger than the third. In music such as waltzes in a *3/4* time signature (three quarter note beats within each measure), the first beat is stronger than the second and third. Through this varied weighting, a multileveled metrical structure arises. Beats that are strong on one level also exist on the level above and therefore reinforce the periodicity of the beat on multiple metrical levels. For example, if we represent beats by dots, a partial metrical structure for a measure in *4/4* time could be constructed as seen in Figure 1.

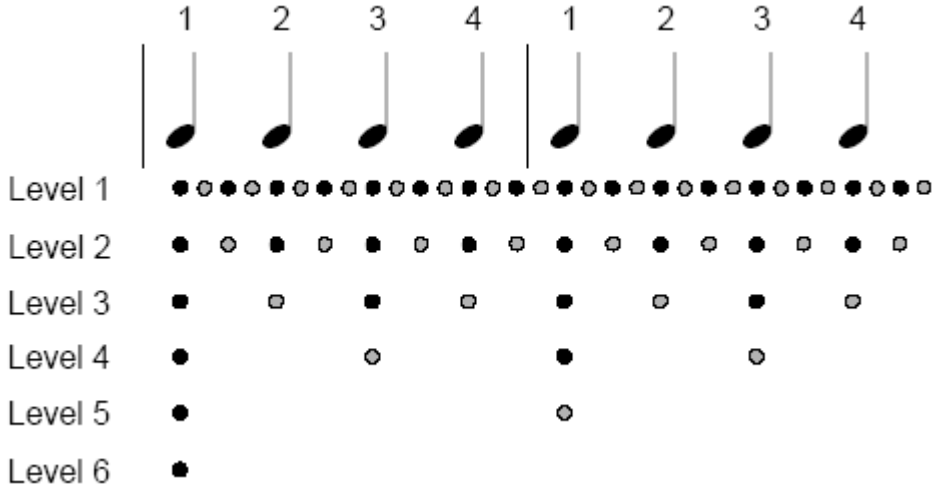

Figure 1: Metrical hierarchy of a four beat bar

Note that, on each level, the odd numbered accents are stronger than the even accents and therefore are also represented on the next higher metrical level. For example, beats one and three in Level 3 are stronger than beats two and four and therefore beats one and three comprise Level 4. This rule applies again to Level 4 in which beat one is stronger than beat three and so Level 5, the metrical level representing the measures or the bars, contains only one accent at the first beat in the measure. A sixth level is shown in the figure in which only the first beat of the first bar is represented. Theoretically, metrical levels can be determined up to the level of the entire piece of music and down beyond the level of the shortest timescale (in this

example, below Level 1, the level representing 16th notes). Moving away from a moderate metrical level (Level 3 or 4 in the previous example) tends to result in a weaker sense of metre for the listener. For example, in a piece with two eight-bar melodic sections, Level 9 would dictate that the first beat of the first section is stronger than the first beat of the second melodic section. This may not necessarily be true or logical and therefore is can be surmised that the metrical hierarchy is a local phenomenon as levels as high as 9, in this case, cease to make logical sense.

It is important to note the periodic relationship between multiple levels of the metrical hierarchy. Referring again to Figure 1, it is evident that each higher level has a period that is twice the period of the previous level. Lerdahl and Jackendoff [5] claim that in Western tonal music, this period ratio is always 2:1 or 3:1. Although this is quite frequently the case, there are many pieces for which these rules of metrical structure do not strictly apply (such as *Paul Desmond's* "Take Five" in which a 5:1 ratio exists or *Soulive's* "One in Seven" in which a 7:1 ratio can be found). In the context of the beat prediction model described here, these exceptions do not greatly affect design decisions but will be examined again in a later chapter.

While there are a limitless number of theoretical metrical levels, listeners tend to focus on one or two intermediate levels. This moderate beat rate is termed the *tactus* [5]. This tactus is the level at which humans are most comfortable following and have the strongest metre perception. It is the level at which a conductor may wave his baton or a listener could tap her foot. In the example shown in Figure 1, this level would most commonly be Level 3. In the remainder of this paper, the term beat will be used to refer to the beats on a possible tactus level of the music. In essence, the *beat* is simply the periodic pulse defined by the metrical level deemed to have to most comfortable tempo by a particular listener.

Studies have shown that humans prefer tempos with periods around 500 ms [6] to 600 ms [7]. This corresponds to tempos of 100 to 120 beats per minute (BPM). Not surprisingly, this tempo range represents the most common tempi found in western music with the vast majority having tempi between 81 and 162 BPM [6]. Listeners are

unlikely to prefer higher metrical levels as the tactus or *beat* since beyond a period of 1500 ms (40 BPM), humans have difficulty tracking the metre as beats begin to seem disconnected [8].

While in theory, the beat is impulse-like, with only location and zero duration, this is not true in practice. When asked, in a personal interview, if all members of a musical group play exactly on the beat, Rod Phillips, a professional musician, held his hands apart at arms length and declared, "The beat is this wide!" What Mr. Phillips was referring to is the fact that musicians often intentionally play slightly ahead or behind the actual beat of the music. This conduct is termed "expressive timing" and is as important as tone, pitch or texture in terms of the feeling and expressiveness in a musician's performance. For example, in a personal interview with experienced drummer and professor Dr. Michael Stone, it was learned that a drummer might play slightly ahead of the beat to add a sense of urgency to the music. Expressive timing is extremely common and creates additional challenges for any beat detection system. The next section begins to address how the proposed research deals with the difficult task of finding this fuzzy notion of the beat.

2.2 Directions for Development

Before moving on, it is worth noting some simple properties of beat detection and prediction systems. In this paper, models that analyze the music and determine the most likely locations of beats already past are termed beat *detection* models. These systems can be either causal, considering only data up to the current point in time, or acausal, making use of data from any time within the performance with the ability to "go back" and change previous beat location guesses. Those that act causally and predict future locations of the beat based on past information are termed beat *prediction* models. Beat detection and prediction systems usually process music in one of two forms: acoustic or symbolic. Acoustic music representations are the analog or digitized forms of the audio signals that humans actually hear. Symbolic music representations, such as MIDI, contain non-acoustic information indicating the location, properties and pitches of the notes that compose the piece of music.

Now that a reasonably formal definition has been given to what is meant by "the beat", a more detailed discussion of the goals and directions of this thesis can occur. Simply put, the goal of this research is to design a system capable of predicting the locations of beats in real-time given an acoustic audio input. In principle, the difference between a beat prediction system and a ***causal*** beat detection system are minor and mostly semantic. Strictly causal beat detection can yield beat predictions by simply using the time of the most recent beat detection and advancing it by the beat period. In a typical prediction system, beats are predicted about one period earlier than they occur giving ample time for another system, auto-accompaniment for example, to process the data. For this reason, once a prediction is made, it cannot be changed until the following beat if input data fails to support the prediction – in other words, the beat prediction is the output of the system, not the beat detection. However, since beat prediction systems also invariably perform beat detection, the terms may, depending on context, sometimes be used interchangeably in the remainder of this thesis.

The process of beat detection is an entirely human perception centred phenomenon. While the proposed beat prediction system is not designed to model human thought processes, it must emulate some of them and be able to produce similar end results. When human listeners find the beat within a piece of music, they extrapolate a metrical grid based on cues from perceived phenomenal accents. The proposed computational system must also use phenomenal accent events to determine a viable beat prediction. Attack points or onsets of notes and sounds within the music represent the most common form of phenomenal accents and will be used to from the basis of the input to the proposed model. The model will attempt to find the best period and location of the beat in the music that aligns itself with the sequence of detected phenomenal accents. Since beats have location but no duration, the output of the proposed system will contain impulses at instances in which a beat is predicted.

Before describing the design and implementation of the proposed system, it is instructive to examine previous models designed by other authors. Many of these other models contain interesting and novel approaches to be detection that may be of use to the proposed design.

Chapter 3: Previous Models

The interesting challenge of creating a computational model for beat detection has been met by many authors over the past twenty years. Over this time period, the advent of more powerful computational devices has enabled the creation of algorithms of greater complexity and the ability to predict and detect the beat in real-time. This trend has culminated in the last five years with scores of new publications on the subject [2-4, 9-17]. This chapter examines a few properties of beat detection systems and some of the more pertinent approaches taken by other authors.

3.1 Classifying Beat Detection Approaches

There is a large variety in the nature of the approaches taken by authors who attempt the difficult task of creating a model for beat detection. However, there are a number of properties that can be examined and contrasted between algorithms that help create a rough classification. One form of distinction, already briefly discussed in the first chapter, involves the nature of the data the system is able to process. Beat detection systems can be classified as working with either acoustic audio data or symbolic data. Systems that process audio data take digitally sampled audio waveforms as input. This input data form is more representative of the sounds that humans use for our own innate beat detection. Systems that fall in this category include those described in [4, 9-11, 16-19]. Symbolic data is usually in the form of lists of note onset times, durations, pitches and other properties. One common example of a symbolic music signal is the Musical Instrument Digital Interface or MIDI. Models that use symbolic data inputs usually restrict themselves to the use of note onset times and include works described in [2, 3, 12, 14, 20-22]. Furthermore, some systems, such as those in [13, 15] can process either digital audio data or symbolic data. The model presented here processes digital audio data.

A second area of classification is distinguishing between process and non-causal models. A process model is one that acts causally, processing the input sequentially [4]. By definition, a real-time system must also be a process system and therefore in order to be considered a candidate for modelling human rhythmic perception, a system must be a process model. Non-causal systems can process the data in any order, examine the entire dataset before making any decisions on the locations of beats and must work off-line. Models described in [10, 11, 13] are all non-causal models whereas those described in [4, 9, 17] and the system proposed in the next chapter are all process models.

Beat detection models invariably contain some degree of musical knowledge or theory. The extent of music theory included in a system can be used as another form of classifier, however, this is clearly a continuous, subjective measure. All models will make use of some rudimentary form of musical knowledge such as the characteristics of a phenomenal accent or defining a metrical grid based on the locations of these accents. Nevertheless, some systems rely to a large degree on higher musical theory. Rule-based systems, such as those created by Povel and Essens [21] and Rosenthal [20] use collections of rules based in music theory to help define the likely locations of beats. Other systems use only a small amount of music theory, such as Large's model [2] that biases tempo detection towards human-preferred tempi. Furthermore, many models use little to no musical knowledge whatsoever, including Scheirer's work [4]. The model proposed in this paper uses very little prior musical knowledge or theory.

The final classification criteria examined here involves the nature and structure of the output of the beat detection or prediction system. Many approaches to beat detection not only attempt to find the beat or pulse in the music, but also construct a hypothesis of the larger metrical structure of the music. This is akin to detecting metrical levels beyond Lerdahl and Jackendoff's *tactus* level thus creating a metrical hierarchy. Models that attempt to find a larger metrical hierarchy include those by Seppänen [9], Parncutt [23], and Rosenthal [20]. Conversely, most models [4, 13, 18, 22] are concerned only with finding the pulse, beat or tempo of the music, including the model presented in the following chapter.

3.2 Pertinent Previous Approaches

There is a wide variety in the methods employed by the designers of systems for beat detection. Many models include some form of rule-based system [20-22], probabilistic system or pattern recognition [9, 14], multiple oscillators [2, 4, 12], as well as systems using techniques such as wavelet analysis [19], self-similarity or autocorrelation [11], and neural networks [15, 24]. A few of the models most pertinent to the proposed system are examined below.

3.2.1 Cariani's Recurrent Timing Network Model

Cariani introduced a basic neural network structure for use in detecting the beat in a musical input in his 2001 paper [15]. This structure, termed the recurrent timing network, is comprised of a tapped delay loop that allows the input signal to be compared to itself with a given modulus. This simple network forms the basis of the beat detection system described in this paper and therefore will be discussed in greater detail in the next chapter.

3.2.2 Desain and Honing's Connectionist Model

One of the earlier beat detection models was proposed in 1989 by Desain and Honing [24]. The purpose of their model was to separate the discrete metrical structure timing from the continuous expressive timing and tempo variations. In order to achieve this, they created a network of connected nodes representing each unique time interval found between note onsets in the input signal. When iterated, the network steers those intervals that are already close to integer multiples of each other to exact integer multiples. This quantization process removes the continuous timing component of the onset timings leaving only the discrete metrical structure.

Desain and Honing recognize that note onsets are likely to occur near the pulse of some level in the metrical hierarchy as described by Lerdahl and Jackendoff [5]. Since all metrical levels are related by integer ratios, steering detected onset time intervals (or equivalently, note durations) has the effect of quantizing note timings onto

the discrete metrical structure, removing any expressive timing. In this way, their model allows interaction, cooperation and mutual adjustment between metrical levels.

3.2.3 Scheirer's Comb Filter Model

Recognizing the potential weakness of beat detection systems that rely on some sort of transcriptive preprocessing of the musical signal from acoustic to symbolic data (such as the common process of onset detection in beat detection systems), Scheirer introduced a system in [4] that detects the pulse of the music directly from the digital audio signal. In this system, comb filters are used on processed versions of the input audio system to detect strong components of the music that occur at a particular characteristic frequency. By using a bank of comb filters with frequencies spread logarithmically from 60 beats per minute (BPM) to 240 BPM, and selecting the filter with the highest output energy, the system is able to determine the approximate tempo of the music.

By detecting the beat directly in the acoustic audio signal, this model does not suffer from a large reliance on the performance of an onset detector preprocessing unit. Moreover, Scheirer's model does not use any specific musical knowledge and seeks only to find strong periodic energies in the input musical signal. This results in a model that is possibly closer to human rhythmic perception and is divergent from those models that are closely related to complex music transcription systems.

3.2.4 Large's Hopf Oscillator Model

Large presents a different approach to finding strong periodic elements in music in his 2000 paper, "On Synchronizing Movements to Music" [2]. Hopf oscillators are used to entrain to regular rhythmic pulsations found in a stream of note onset impulses. A network of these oscillators tuned to periods meant to span the range of beat perception, distributed logarithmically with periods between 100 ms and 1500 ms create self-sustained oscillations in response to elements within the input with matching characteristic frequency. The energy level of a particular oscillator grows when its pulsation correlates well with the locations of impulses in the input. Coupling between

the oscillators is created such that they may compete for activation. In this way, oscillators that best represent the period of the input signal should tend to inhibit those that do not. The oscillators that maintain high activation are used to describe the rhythmic structure, period and phase, of the input signal.

The use of coupling between potential beat period hypotheses provides a behaviour not seen in Scheirer's model but which is relevant to human beat perception. In this way, two compatible period hypotheses that represent different metrical levels of the same rhythmic interpretation can reinforce each other. Equally, hypotheses that conflict inhibit each other. Large accomplishes this by creating small inhibition between oscillators with periods that are harmonic and sub-harmonic multiples of 2 and 3, and large inhibition with those that are not. This allows a more intelligent selection of the beat hypothesis that best represents the input signal by weighing all available information.

3.2.5 Seppänen's Pattern Recognition and Tatum Model

A recent Master's thesis by Jarno Seppänen [9] proposes yet another approach to causal, real-time beat detection using a probabilistic pattern recognition methodology. Seppänen's approach involves first finding the *"tatum"* metrical level – that is, the highest metrical level of which all onsets fall on the pulse. The tatum is used to define the lowest metrical level above which higher levels with integer multiple periods can be found. One of these higher levels is eventually determined to represent the actual beat in the music stimulus. This process involves first applying a phenomenal accent model to detected onsets. Then, using this stream of accents, the system can determine where on the metrical hierarchy the beat is most likely to occur.

One of the interesting features of this model is its ability to handle small tempo fluctuations and slow tempo changes. Unlike the Scheirer or Large models, a small change in the tempo of the input music need not result in the selection of a new beat period prediction. Seppänen's model can slowly adjust the period of the tatum while allowing the highest metrical structures to remain intact. In this manner, the detected beat is slowly altered with the changing tatum period and no hypothesis re-evaluation is

required. Furthermore, unlike the Large and Scheirer models, Seppänen's method allows for precise beat period matching.

3.3 Further Directions

Each of the aforementioned models contains an interesting or novel aspect that could contribute to a more robust, better performing beat detection system. Desain and Honing's model [24] contains cooperation between neurons to steer detected metrical levels toward integer multiples of each other. Scheirer's model [4] works without first detecting onsets (unlike the Desain and Honing, Large, and Seppänen models), claiming better performance by skipping this error-prone step. Large's model [2] uses an aspect of inhibition between competing beat hypotheses – hypotheses that are all being evaluated simultaneously. Finally, Seppänen's model [9] allows for precise tracking of tempo changes without beat prediction re-evaluation. Many of these concepts will be considered in the design of the model proposed in the next two chapters.

Chapter 4: Recurrent Timing Networks

Every beat detection system must be able to infer a regular metrical structure from the input musical signal. Usually, this amounts to finding periodicity or hidden structure within some perceptually motivated signal or time-series. Chapter 2 discussed how beat detection involves determining this metrical structure from a perceived stream of phenomenal accents. In the proposed model, the locations of phenomenal accents are derived from note and sound onsets in the music. The simple structure responsible for finding periodicity within this stream of accents is the *Recurrent Timing Network* (RTN), first proposed for use in beat detection by Cariani [15]. Cariani's original implementation of recurrent timing networks will first be examined, followed by a discussion of the changes and enhancements made for use in the proposed system.

4.1 Cariani's Recurrent Timing Networks

The application of recurrent timing networks to musical beat detection was first discussed by Cariani in a 1999 working paper [25] and then in more detail two years later [15]. While originally proposed for use in cognitive modelling, recurrent timing networks provide a computationally efficient method of periodic pattern detection.

A recurrent timing network is a simple neural network that allows a signal to be compared to itself at specific instances in the past. It is constructed using a tapped delay loop that feeds back upon itself, permitting the input signal to affect recurring network activation levels. These activation levels propagate around the loop, from neuron to neuron, advancing once per time-step, and recur with a fixed period equal to the number of neurons in the network. Figure 2 shows a recurrent timing network with a delay length, or period, of four.

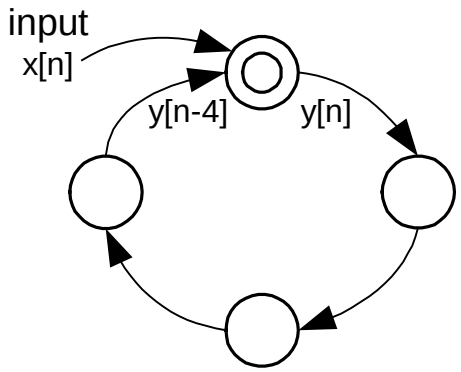

Figure 2: A recurrent timing network of length four

The node at which the loop recurs upon itself and receives new input is termed the *head node*. At each time-step, input is presented to the head node together with the activation level from the last node in the loop. For the sake of this discussion, the input, $x[n]$, will be simply defined as a time sequence of values lying in the range [0,1]. The input and the recurrent activation level are used to calculate a new activation level for the head node, which, at the next time-step, will be propagated to the second node in the network. As such, the activation at the head node at time-step n, $y[n]$, is a function of the current input value, $x[n]$, and all input values at periodic time-steps in the past, $x[n-k \cdot p]$, where k is an integer and p is the period of the network. Expressed more succinctly, the activation at the head node can be expressed as a recursive function as shown in Equation 1.

$$y[n] = \mathrm{F}\big(x[n], y[n-p]\big)$$

Equation 1

The activation levels at all other neurons in the network are also recursive functions of the input and previous activations, but in this case, with phase shifts ranging from *1* to *p-1* time-steps. In this manner, the recurrent timing network simultaneously maintains activation levels corresponding to periodic recursive functions of the input at all possible phases.

By customizing this recursive periodic function, one can implement a large number of different algorithms. For example, using the function shown in Equation 2,

a recurrent timing network could emulate Scheirer's comb filters [4] with gain α. It is evident that this simple neural network structure can be used for implementing a variety of algorithms useful to recurrent pattern detection.

$$y[n] = \alpha \cdot y[n-p] + (1-\alpha) \cdot x[n]$$

Equation 2

Cariani suggests a simple geometric growth recursive function for use in his recurrent timing network implementation, designed to process input time sequences, $x[n]$, consisting of ones and zeros. The existence of a one in the input signal corresponds to the existence of a phenomenal accent in the acoustic music signal under consideration. When a nonzero input is coincident with an existing nonzero activation level propagating to the head node, the resulting activation should be reinforced with a facilitation factor of β. If, however, the input is equal to zero, activation in the head node is reset to zero. If there is no activation propagating to the head node in the presence of nonzero input, the activation level becomes equal to the input. This function can be expressed as Equation 3.

$$y[n] = \max\{\beta \cdot x[n] \cdot y[n-p], x[n]\}$$

Equation 3

Through the use of the head node function shown in Equation 3, if a pattern that is fed into the network matches a pre-existing pattern, it is reinforced within the loop. If the new pattern is different, old activations are cleared and the new pattern is set into the network. Therefore, elements within the input that are periodic with the same period, p, as the network, cause neuron activations to grow. As Cariani states, "an incoming pattern becomes its own matched filter pattern-detector" [15]. Consequently, recurrent timing networks allow hidden patterns to be found in a stream of complex inputs.

There are a variety of inherent strengths in the design of the recurrent timing network. The most important strength lies in the network's ability to simultaneously test periodicities in all possible phases in a computationally efficient manner.

Furthermore, not only can the network find strong periodic elements within the input signal, it can also detect strong periodic patterns. This allows the network to be able to detect more than just the beat within a stream of phenomenal accents – it can also find other elements such as dominant rhythmic patterns, the location of the offbeat, or the existence of a swing or shuffle beat.

Finally, while the network is extremely useful for detection, it can also be used for periodic element, or beat, prediction. In fact, this behaviour already exists in the network and requires no extra logic. At every time-step, the head node receives an old activation level from the last node in the network and a new external input value. This incoming activation level can be used as a form of prediction for the external input. A high incoming activation level indicates that there has been strong tendency in the past for a nonzero input to occur at this phase in the network's period. Larger incoming activations imply stronger evidence of this past behaviour and a higher confidence that this trend will continue in the future. In this way, incoming activation levels act as predictors for the presence of coinciding input pulses. If the input pulse does indeed occur, the expectation is strengthened for next time through an increase in the activation level. If the pulse fails to occur, the prediction may have been wrong and the activation is weakened. Thus, activation levels propagating to the head node act as a prediction indicator of the periodic element they represent.

While the recurrent timing network implementation used by Cariani has many strengths, Cariani also recognizes its many weaknesses [15]. The most prevalent weakness in Cariani's implementation is that large activations and strong recurrent patterns within the network disappear after only one missed cycle. This is an extremely undesirable behaviour, especially when trying to detect a beat in a series of onsets since no assumption can be made that an onset will fall on every beat. A serious consequence of this weakness is that it becomes unlikely that strong periodic patterns can be built given anything other than a trivial input signal. Moreover, activations in the network cannot be relied upon to accurately represent the strength of a recurrent pattern in the input signal.

A second weakness stems from the discrete-timing nature of the network. Periodic patterns that are slightly non-isochronous, have non-integer periods, or that suffer from pattern jitter cannot be properly detected by simple recurrent timing networks. A third and final weakness of the recurrent timing network is the inability to properly track changing tempi. Most of the aforementioned weaknesses of the recurrent timing network structure will be accounted for and corrected in the proposed system. The next section describes a possible solution to the first problem – decaying network activations.

4.2 Improvements to Cariani's Recurrent Timing Networks

Three major weaknesses in Cariani's recurrent timing network implementation were identified in the previous section: non-robustness to missed cycles, sensitivity to pattern jitter, and inability to track tempo changes. The first weakness will be addressed here in the form of modification to the activation level calculation function in the head node.

Before continuing, it is worth clarifying the nature of the input signal to the network. So far, the input signal, $x[n]$, has been defined only as a time sequence of values in the range [0,1]. A value of zero in the input corresponds to a lack of input stimulus, whereas a nonzero value indicates the presence of an input stimulus of strength proportional to said value. While this definition of the input signal is still quite vague, it is sufficient for the current discussion and will be further discussed in the next chapter.

There are five requirements and desired behaviours that should be imposed upon a successful recursive periodic head node, characteristic function:

1. **Reinforcement Causes Growth:** Pattern reinforcement in the form of a strong input value that is coincident with a large recurrent activation level should cause growth in the activation level. This growth should be proportional to the strength of the input. This is a fundamental aspect of the recurrent timing

network and is present in any well-designed characteristic recursive function. Cariani's model already exhibits this behaviour.

2. **Absence of Reinforcement Causes Decay:** The activation level entering the head node should be decayed if the input is found not to support a strong activation. Through this behaviour, a prediction arising from a strong detected periodicity in the input signal should be weakened by the lack of corroborating input. Cariani's model exhibited an exaggerated form of this behaviour where activations were zeroed in the absence of a non-zero input value. Again, this is one of the major weaknesses of Cariani's system. Ideal behaviour would permit repeated missing supporting input values to result in a steady decay of activation levels, eventually reaching a level indicating the lack of any input signal periodicity.

3. **Activation Levels Should be Bounded:** An upper and lower bound should be placed on the neuron activation levels in the recurrent network. An example best explains the justification for this requirement. If, over the course of one minute, a pattern in one timing network were to be reinforced 100 times and a pattern in another network were to be reinforced 150 times, there should be little to no difference in the activation levels of the two patterns. Both patterns have received ample supporting evidence and should reside at a level indicating maximum pattern strength. Any activation that reaches this maximum level should be considered very strong and its strength need not be distinguished from other strong activations. Accordingly, a lower bound on the activation level should be instated that simply indicates the complete absence of evidence of a recurrent pattern.

4. **Noise Suppression:** There should be some form of noise tolerance, both at high and low activation levels. At very low levels, no recurrent pattern exists and so the arrival of two or three periodically coincident inputs should not cause a large growth in activation. Such a small string of periodic elements is not sufficient to indicate the existence of a strong periodic element and therefore

can be considered noise in need of suppression. The inverse case applies to high activation levels and an occasional missing input. If a neuron's activation is high, indicating a strong periodicity, one or two missing reinforcing inputs should not greatly reduce the activation level. A large decay in such a circumstance could cause the affected pattern to appear to be significantly weaker than other strong patterns when, in fact, this is not the case. One or two missing inputs coincident with high activations can also be considered noise and should be suppressed.

5. **Recursive Function:** For purely pragmatic implementation-related reasons, the selected characteristic function must be a recursive function of the input and past activation levels. This is a necessary condition, required to fit the function into the existing framework of the recurrent timing network without the need for additional variables or storage elements.

One function that satisfies all of the above requirements is a discrete recursive function resembling the sigmoid function commonly found in neural network texts [26]. Equation 4 gives the form of this proposed sigmoid characteristic function where $x[n]$ is the external input, $y[n\text{-}p]$ is the activation level recurring in the network loop to the head node, and $y[n]$ is the new head node activation level.

$$y[n] = y[n-p] + x[n] \cdot y[n-p] \cdot \left(1 - y[n-p]\right)$$

Equation 4

In this implementation, the input, $x[n]$, must be scaled from the range [0,1] to a span of [-0.5, 1], where negative values cause decay resulting from missing input values, and the range (0,1] is used for nonzero input values of varying intensities. The output of activation level, $y[n]$, is bounded between γ and $1\text{-}\gamma$, where γ is a minimum activation bound. Smaller values of γ result in more noise suppression at the low and high ends of the function. Figure 3 shows the activation level generated by this function under continuous reinforcing input as defined in Equation 5, with γ equal to $1/32$. In Equation 5, p is the recurrent timing network period.

$$x[n] = \begin{cases} 1, & n = k \cdot p, \quad k \in I \\ 0, & otherwise \end{cases}$$

Equation 5

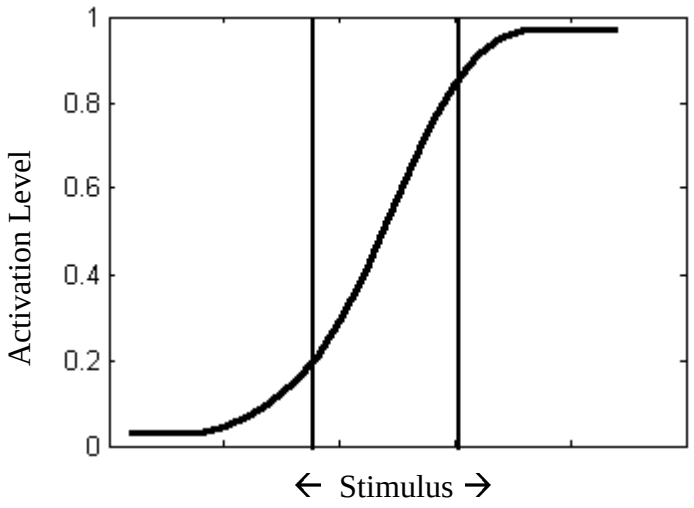

Figure 3: Recurrent timing network activation level behaviour

Note that the activation level grows only slowly at first, the positive noise suppression stage, followed by rapid growth, the activation reinforcement stage, and finally grows slowly again, the negative noise suppression stage. These regions are delineated with the thick vertical lines shown in the figure. Under reinforcing and inhibiting input values, the activation level moves approximately up and down (right and left) along the curve, respectively.

The proposed characteristic function given in Equation 4 and shown in Figure 3 satisfies all of the five required behaviours. If a low activation level receives occasional positive input values, the level will remain low, suppressing this purported noise. If a high activation level were to incur occasional negative input values, this form of noise would also be suppressed and the activation would remain relatively unchanged. Only under repeated positive or negative inputs will the function tend to grow or decay by any substantial amount. Therefore, the proposed characteristic function is an ideal candidate for use in the recurrent timing networks used in the proposed beat detection and prediction system.

4.3 Cariani's Model

It is worth briefly making mention of the remainder of Cariani's beat detection model described in [25]. Cariani's system does not involve much beyond simple analysis of musical signals using recurrent timing networks. Networks ranging in period from 0.01 seconds to 2 seconds in 0.01 second increments were used to detect periodic elements in a musical stimulus consisting of either sequences of zeros and ones or scaled audio signal amplitude envelopes. By examining the mean and standard deviation of the activation levels in each loop, Cariani was able to subjectively select the loops with the strongest recurrent patterns. Visual inspection of the contents of these loops showed the robust detection of patterns within the input, but no attempt was made to create an automated system for beat detection outside of the recurrent timing networks themselves.

The model proposed in the next chapter seeks to use these basic neural structures to form the basis of a robust and complete beat prediction system.

Chapter 5: The Proposed Model

While recurrent timing networks are able to perform basic periodicity detection and will comprise the basis of the detection and prediction logic in the proposed system, they alone are not sufficient for robust beat prediction. The beat prediction model proposed here is a fully automated solution that is able to process digital audio signals and create an output consisting of impulses at times that beats are predicted.

This new model uses a multi-agent approach to beat detection and prediction. In essence, this means that the system creates a set of possible beat periods and assigns one such hypothesis to each agent. Every agent processes the input phenomenal accent stream and tracks the performance of its hypothesis using a recurrent timing network tuned to the agent's period hypothesis. The system then selects the agent containing the beat hypothesis that appears to be the strongest or most representative of the actual beat in the input music. This agent is used to generate the predictions of future beat locations.

The proposed system consists of three main units. The first unit, an onset detection unit, processes the incoming digital audio data and creates the stream of phenomenal accent note onsets. The second section performs the inter-onset time interval statistic collection needed to generate possible beat period hypotheses from the onset stream. The third and final unit contains a pool of beat detection agents that attempt to find the beat in the onset stream and compete for selection as the best agent.

Figure 4 shows a block diagram of the entire system. In the following sections, the design of each component will be discussed in detail. The final section in this chapter will provide brief implementation information.

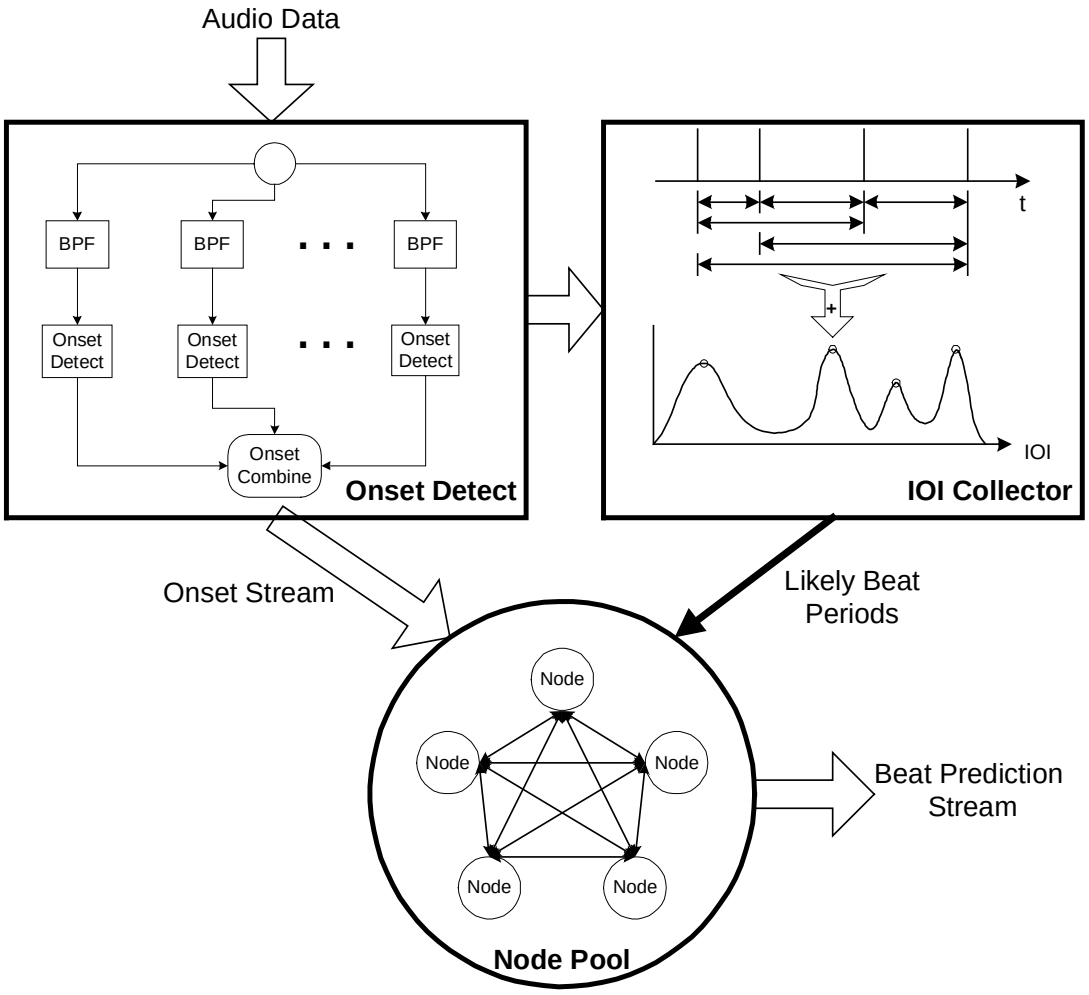

Figure 4: Beat prediction system block diagram showing main components

5.1 Onset Detection

It has been stated many times that the proposed beat detection system attempts to infer the position of the beat from the locations of phenomenal accents in real audio data. In order to do this, the locations and properties of these phenomenal accents events must be extracted from the digital audio signal. The form of phenomenal accent used in this model is the onset of a sound or a note. Consequently, this first stage must detect these sound onsets in the original audio data. This stage essentially translates real acoustic audio data into a symbolic form that the remaining parts of the model are able to use for beat detection and prediction.

The purpose of this research is to design a new approach for the detection and prediction of the beat in an acoustic audio signal. While onset detection is a crucial part of this system, it is a research area in and of itself and is not the focus of this work. This onset detection process used is not novel to this research and is based on the works of Sheirer [4], Seppänen [9], Klapuri [27], and Duxbury et al [28].

This onset detection process can also form one of the first stages of most automatic music transcription systems – the stage responsible for finding note, chord, and sound locations. While much progress has been made in the field of automatic transcription, a successful solution has not yet been found [4]. For this reason, Scheirer [4] criticizes beat detection systems that employ this "transcriptive" approach involving first detecting note onsets. Not only is the onset detection stage likely to introduce error into the system, but also it is unlikely that this approach is akin to the way in which humans perform beat detection. The system described in this chapter attempts to be a robust beat detection system that should have low sensitivity to errors introduced in the onset detection stage. In fact, spurious and missing onsets are acceptable since the proposed model should be able to find a hidden beat in a complex or incomplete input. Furthermore, the proposed system does not purport to model human beat prediction. While the human brain is an excellent beat detection system, it does not necessarily employ the only successful detection method.

5.1.1 Theory of Onset Detection

The goal of an onset detection system is to identify the location and strength of sound or note onsets within the input musical signal. Lerdahl and Jackendoff define the corresponding phenomenal accent as "attack points of pitch events" or "sudden changes in dynamics or timbre" [5]. In order to detect these psychoacoustic phenomena, we examine the work of Scheirer [4].

The detection of note locations and their corresponding onsets can be extremely complex and prone to poor performance. However, there are a number of signal manipulations and simplifications that are able to maintain the rhythmic perception found in the original music. Scheirer found that dividing the input signal into a number

of separate frequency bands and extracting the signal amplitude envelope for each band provided sufficient information to preserve the signal's original sense of pulse and metre. Other information, such as the notes themselves, is unimportant to the rhythmic structure.

From Scheirer's work it can be determined that the amplitude envelopes of each frequency band should be processed separately and results should be combined only after processing. While this level of simplification greatly reduces the amount of signal information that requires processing and is sufficient for recovering the beat, it is also necessary, as additional simplifications do not always preserve the beat. Examples of inappropriate simplifications are using only one large frequency band or linearly combining amplitude envelopes before processing.

The design of the onset detector from the proposed model makes use of these findings by Scheirer. The input audio stream is first divided into eight non-overlapping frequency bands. An onset detection scheme is then applied to the signal amplitude envelope of each band independently. Finally, the detected onsets from each band are combined to form a single onset stream. This process is illustrated in Figure 5. Each step of this detection system will be covered in more detail in the following sections.

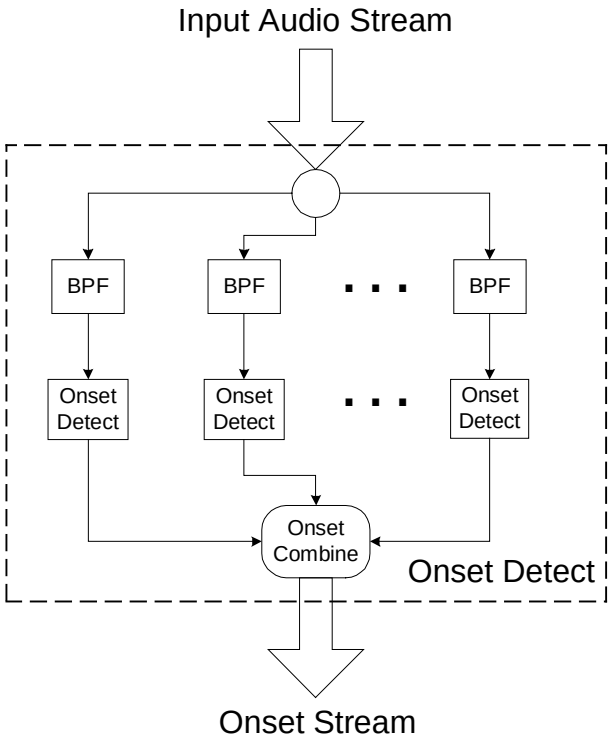

Figure 5: Overview of the onset detection process

5.1.2 Frequency Band Separation

Scheirer suggests that a minimum of four separate frequency bands is required to maintain the percept of rhythm within the simplified signal. Many authors employ this frequency separation technique for onset detection and the number of frequency bands selected almost always differs. Some authors use as few as five bands [28], and others use as many as 25 bands [15]. Empirical studies have shown that the selection and the nature of these bands does not significantly affect onset detection system performance [4]. Eight frequency bands, logarithmically spaced, were selected for use in this onset detection unit, where every band, except the lowest and highest, encompasses exactly one octave.

The required filters were implemented as sixth-order Butterworth infinite impulse response (IIR) filters designed using Matlab's filter design toolbox. Table 1 gives the filter parameters and average group delays for the filter assigned to each band. Figure 6 shows the frequency response of all eight filters. In order to synchronize

detected onset streams from all bands, the filtered signal from each band is delayed an additional amount such that the total delay including the inherent filter group delay is equal to 6.848 ms. At this point, the total system delay is therefore 6.848 ms.

Table 1: Bandpass filter parameters and properties

Band Number	Cut-off Range (Hz)	Bandwidth (Hz)	Average Group Delay
1	0 - 110	110	6.825 ms
2	110 - 220	110	6.848 ms
3	220 - 440	220	3.424 ms
4	440 - 880	440	1.701 ms
5	880 - 1760	880	0.862 ms
6	1760 - 3520	1760	0.431 ms
7	3520 - 7040	3520	0.204 ms
8	7040 - 14080	7040	0.113 ms

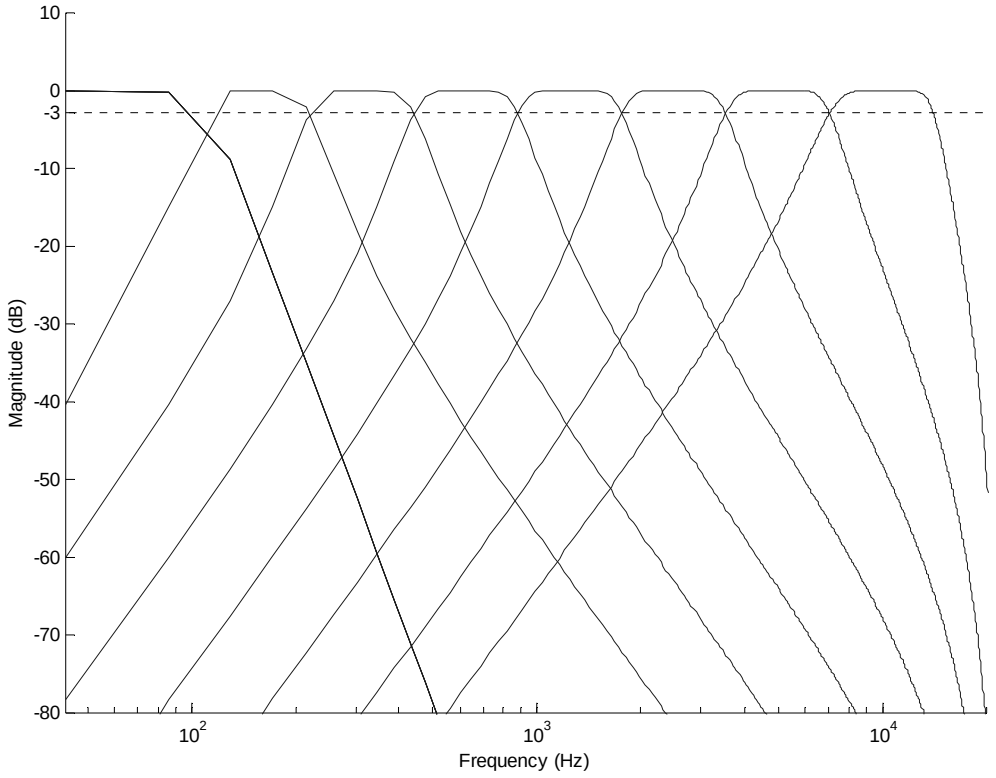

Figure 6: Frequency magnitude response of onset detector filter bank

5.1.3 Band-wise Onset Detection

With the input digital audio signal separated into its component octaves, an identical onset detection unit is applied to each frequency band. Onset detection in audio signals is akin to edge detection in images – rapid changes in intensity mark onsets or edges. In fact, many approaches used for edge detection in image processing have been applied to sound onset detection, including peak detection in the first-difference and multi-scale edge detection [29]. The band-wise onset detection approach used here is based on Seppänen's first-difference approach [9]. Seppänen's system is selected due to its reliable onset detection performance and because it is based on the successful onset detection work presented by Klapuri [27].

The first step in the selected onset detection scheme is the discovery of the band-limited signal amplitude envelope. In [9], Seppänen calculates this envelope by taking a low-pass filtered root mean square (RMS) estimation of the signal. This process is shown in Equation 6, where *x[n]* is the input signal, *y[n]* is the RMS output signal and *h[n]* is the impulse response of a low-pass filter with a cut-off frequency of 30 Hz. This low-pass filter causes an additional 8.993 ms of mean group delay to each band, increasing the total delay of the system to 15.84 ms.

$$y[n] = \sqrt{x[n]^2 * h[n]}$$

Equation 6

This RMS signal provides a good estimate of short-time power of the original signal, *x[n]* [9]. The resulting signal, *y[n]*, is decimated by a factor of 100 to ease the computational burden on the remainder of the algorithm. This decimation has little to no effect on the onset detection viability of the signal since the vast majority of the signal energy is well below the new Nyquist frequency. The next step involves convolving the decimated signal with a 100 ms Half-Hanning (raised-cosine) window. This performs energy integration, preserving rapid amplitude changes but masking rapid modulation much in the same way as the human auditory system [27, 29].

The final step in the band-wise onset detection component detects rapid changes in the resulting signal energy envelope. To do this, Klapuri applies the concepts of auditory just-noticeable-difference (JND) and the Weber fraction to this problem. He notes that the smallest detectable change in sound intensity, ΔI, is proportional to the signal intensity, I – therefore, $\Delta I/I$ is approximately constant. Since sound intensity is proportional to the square of the amplitude, the intensity can be replaced with the square of the local signal mean, and the change in intensity can be represented by the first-difference of the squared signal [9]. Equation 7 shows this relationship.

$$\frac{\Delta I}{I} \propto \frac{y^2[n] - y^2[n-1]}{\left[\left(y[n] + y[n-1]\right)/2\right]^2}$$

Equation 7

Where $y[n]$ is the signal energy envelope as calculated above. An onset can be considered to have occurred when this just-noticeable-difference rises above a psychoacoustically predetermined threshold, κ. This threshold is set by Seppänen to equal to $10^{-6/10}$ [9]. By expanding the difference of squares, the expression dictating when an onset should be detected, shown in Equation 8, can be found.

$$b[n] = \frac{y[n] - y[n-1]}{y[n] + y[n-1]} \geq \frac{\kappa}{4} \approx 0.06$$

Equation 8

Equation 8 gives the threshold function used by Seppänen to detect onsets. While this function is reasonable it fails to detect onsets with anything less than a very strong attack, such as a drum hit. The problem lies in Equation 8's use of only two consecutive samples to approximate the local energy slope. To better approximate this value, an expression taken from the works of Duxbury et al can be applied [28]. Equation 9 shows the calculation of a new threshold function. This new function is simply a weighted sum of K versions of Equation 8, permitting a better approximation of the local slope that can be used to detect onsets more robustly. In this implementation, K was selected to equal 10.

$$\tilde{b}[n] = C \cdot \sum_{i=1}^{K} \frac{1}{i} \cdot \frac{y[n] - y[n-i]}{y[n] + y[n-i]} \geq 0.06$$

where,

$$C = \sum_{i=1}^{K} \frac{1}{i}$$

Equation 9

When $\tilde{b}[n]$ exceeds 0.06 an onset should be detected. Another onset cannot be detected until a minimum duration of 100 ms has passed, or the function $\tilde{b}[n]$ drops below a threshold of -0.035 [9]. This ensures that only one onset is detected per rise in the signal energy envelope.

Onset intensity or strength is also calculated using the signal energy envelope. The time-step, n_a, at which an onset is detected is termed the onset attack time. The time-step, n_m, at which the signal energy envelope reaches a local maximum following the attack time is termed the maximum amplitude time. The onset intensity is then found as the difference between the energy amplitude at the maximum and the attack point, or $y[n_m]$-$y[n_a]$. Since the energy envelope, $y[n]$, lies between zero and one, the calculated onset intensity is also scaled between zero and one. The output onset stream from this unit contains zeros where no onset is detected and nonzero onset intensity values at the attack points, n_a, where onsets are detected.

5.1.4 Combining the Onset Streams

The final stage in the onset detection unit of the proposed beat prediction system involves combining the onset streams from each frequency band's detection unit to form a single onset stream for the entire input audio signal. It is common for a strong onset to be detected in many frequency bands, however it is not usually detected at precisely the same instant. For this reason, the combination process must select the most appropriate onsets while ignoring all others.

Onset combination processes used by other authors range from the complex [27], to the simple [9, 28]. The method used here is simple and is similar to the

processes used by Seppänen [9] and Duxbury et al [28]. Both authors combine the onset streams through addition followed by a simple filtering scheme to remove duplicate onsets. While Duxbury et al suggest a weighting scheme to give onsets in the higher frequency bands precedence, the selected approach is closer to Seppänen's in which all bands are considered equally. The first stage of combination can be expressed using Equation 10 where $o_k[n]$ is the onset stream from the k^{th} frequency band.

$$o[n] = o_1[n] + o_2[n] + ... + o_8[n]$$

Equation 10

The second step involves removing duplicate representations of the same onset from the compiled stream. An approach similar to Duxbury et al is employed in which a 50 ms window is slid across the onset stream, allowing only the onset with the greatest intensity to remain. The effective result of this operation is that all remaining onsets are a minimum of 50 ms apart. This windowing function requires the addition of a 50 ms signal delay, bring the total system lag to 65.84 ms. The processed onset stream is then ready to be used in the remainder of the proposed system.

5.2 Inter-Onset Interval Statistic Collector

The onset detection stage acts as a preprocessor for the input audio data, converting it to a symbolic form that can be processed by the rest of the beat prediction system. The form of this symbolic data is more signal than symbolic but it does contain the locations and strengths of the phenomenal accent sound onsets detected in the original audio data. The next component in the beat prediction system determines the set of most likely beat periods given this stream of onsets. The set of periods is used to generate beat period hypothesis agents in the final stage of the system.

The crucial idea that finding the beat in a piece of music involves determining the metrical structure using cues from phenomenal accent locations has been reiterated many times. One of the important ways in which this knowledge is useful is that it means that these phenomenal accents frequently land on the beat. From this, it can be

deduced that the time interval between two onsets that lie on the beat corresponds to the beat period. Since a large number of onsets are coincident with the beat, an equally large number of the intervals between onsets are equal to the beat period.

In order to determine the likely candidates for the beat period in the music under consideration, a strategy similar to that of Desain and Honing [24] is employed. Inter-onset intervals (IOIs), or the time difference between two detected sound onsets is used to determine the likely beat period candidates. Contrary to the conventional definition, an interval does not have to be exist between successive onsets, but can be found between any two onsets in the input stream [13, 20]. Figure 7 shows a set of four onsets on a timeline with the six possible inter-onset intervals measured beneath. Onsets are shown as impulses and intervals are labelled 'a' through 'f'.

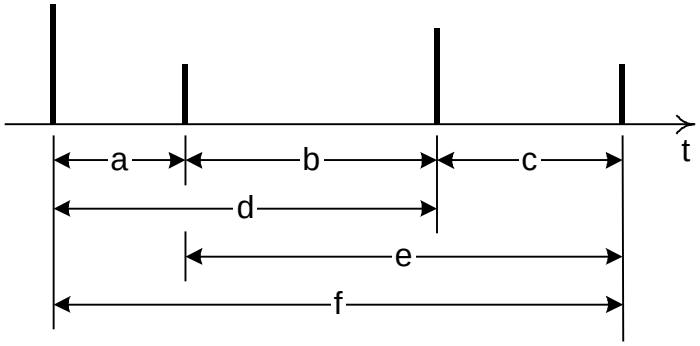

Figure 7: Four onsets with inter-onset intervals shown below

Notice that the most recent onset has intervals measured to the three previous onsets, labelled 'c', 'e' and 'f'. In the IOI statistic collector described here, intervals are measured up to a maximum length of 1.2 seconds. This corresponds to a minimum detectable beat tempo hypothesis of 50 beats per minute – a reasonable lower detection limit. To find the most frequently occurring inter-onset intervals, a histogram depicting the frequency of detection is created.

5.2.1 Determination of IOI Frequency of Occurrence

Finding the most common inter-onset intervals in an onset stream is akin to approximating the distribution of a random variable corresponding to inter-onset

interval probability distribution. There can be no assumption of precise onset time detection nor can accurate timing by the musicians be assumed in the original musical input. For this reason, the measured inter-onset intervals may deviate significantly from their perceived or intended metrical timing. Since onset times are frequently normally distributed about their mechanical means [14], a reasonable assumption can be made that detected inter-onset intervals are also distributed normally about their intended duration. As a result, an effective method for creating an accurate representation of inter-onset interval frequency of occurrence distribution is to use the Parzen windowing technique commonly used in non-parametric probability density estimation. In this case, for each measured inter-onset interval a Gaussian window is added to the current distribution estimate. This is more accurately expressed in Equation 11, where a fill function for a continuous dominant IOI distribution is given. In this equation, $W(t)$ is the Gaussian Parzen window function with variance equal to five input signal sample periods, $f_t(k)$ is the fill function at timestep t, and IOI_j is the j^{th} of J inter-onset intervals to be added in this timestep.

$$f_t(k) = \sum_{j \in J} W\left(k - IOI_j\right)$$

Equation 11

Inter-onset intervals to a maximum of 1.2 seconds are tracked in the dominant IOI distribution estimate. The purpose of this distribution is to represent the frequency of occurrence of each possible interval detected in the music. It is important that it contains relevant, meaningful data, and therefore a sufficient number of inter-onset intervals must be accumulated. If too few intervals are collected, the distribution cannot be assumed to properly represent the interval occurrence frequency. Conversely, it is also important that the distribution be responsive to changing statistics within the input music signal. For this reason, intervals collected far in the past should have a minimal effect on the present dataset. A balance must be found between these conflicting requirements.

Seppänen proposes a solution to this conflict by employing a time-varying approach to the distribution [30]. Although Seppänen employs only a simple IOI histogram for reasons divergent from the use intended here, both applications require representative IOI distribution data. To enable recent data-biased time-varying distribution behaviour, a leaky integrator system is used in which values exponentially decay with the passage of time. Modification to the distribution at timestep *t*, $h_t(k)$, follows Equation 12, a method similar to that given in [30].

$$h_t(k) = d \cdot h_{t-1}(k) + f_t(k)$$
$$d = 0.5^{(t_u / t_{hl})}$$

Equation 12

The parameters in Equation 12 are as follows: $h_t(k)$ is the updated value of the distribution at timestep *t* for interval *k*, $f_t(k)$ is the fill function at timestep *t* for interval *k*, and *d* is the decay parameter. The decay is calculated as a function of the time since the last distribution update, t_u, and the histogram half-life time, t_{hl}. By using this decay function, the distribution decays at a constant rate over time.

Using the Parzen windowing and distribution decay techniques has the effect of creating a smooth representation of inter-onset interval distribution and aids in better detection of dominant interval peaks. Figure 8 depicts a typical interval distribution given realistic musical input data. Dominant peaks appear to be easily identifiable.

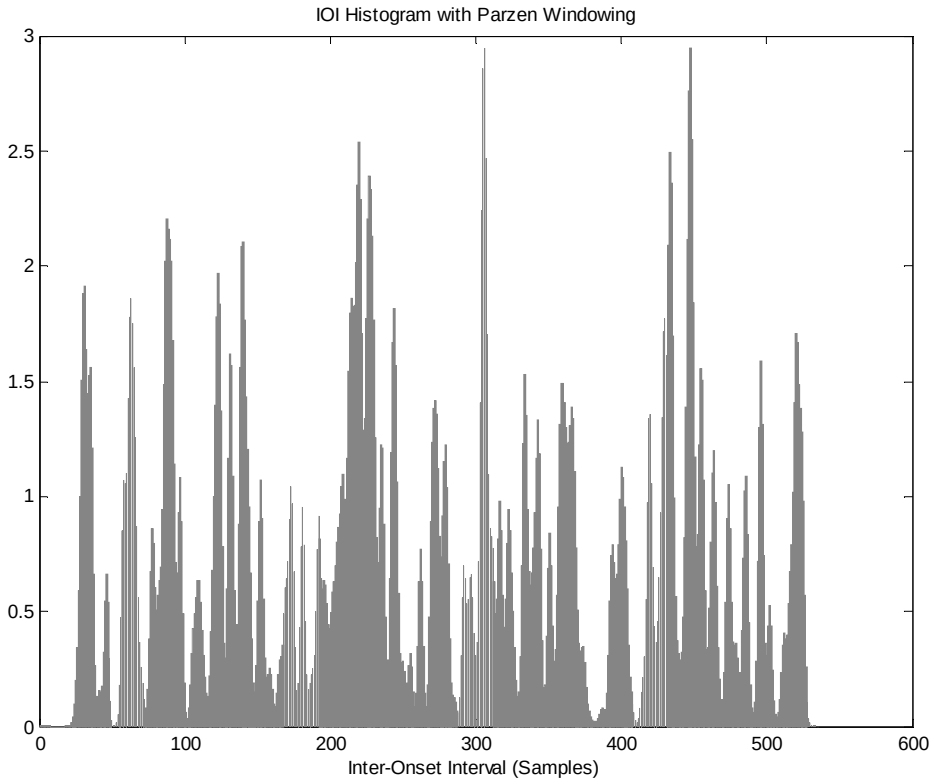

Figure 8: IOI histogram with Parzen windowing

5.2.2 Detecting Likely Beat Periods

The purpose of the inter-onset interval statistic collector is the determination of likely beat periods in the given music input. These beat periods are found by locating peaks in the IOI occurrence distribution and are used to create beat hypotheses for the independent agents found in the final stage of the system. If the distribution under consideration appears similar to that shown in Figure 8, this task is not difficult.

Before discussing the detection of peaks in the distribution, a significant implementation detail must be addressed. As the proposed beat prediction system must be implemented in a discrete digital environment, the use of continuous IOI distributions is not feasible. For this reason, the implementation of the Parzen windowing method used to create the IOI occurrence distribution must be handled in a discrete fashion. Consequently, the IOI distribution is discretized by sampling at a

resolution of one sample per input signal sample. In this manner, the IOI distribution can be viewed as a well-behaved, discrete histogram, $h[k]$, from this point forward.

There are two conditions that must be met for an interval, k, in the distribution to be considered a peak. First, the distribution level, $h[k]$, must be high relative to the rest of the levels in the distribution – it must be dominant. This is accomplished by only considering intervals whose level exceeds a preset threshold level. This threshold can be calculated using Equation 13.

$$T = \alpha \cdot \text{mean}\{h[k]\} + (1-\alpha) \cdot \max\{h[k]\}$$

Equation 13

The parameter α in Equation 13 is a mixing coefficient in the range [0,1] and is selected, in this implementation, to equal 0.75. For interval k to qualify for selection as a dominant peak, $h[k]$ must be greater or equal to T. The second condition for selection as a dominant peak requires that the interval under consideration is a local maximum. This is tested using Equation 14. If an interval, k, meets both conditions of sufficient level and local maximality, it is considered a dominant peak.

$$h[k] > h[k+i], \quad i = -2,-1,1,2$$

Equation 14

Assuming that the local maximum occurs exactly in the centre of an interval bin meeting the above requirements is not entirely valid. The distribution around the detected local maximum could be heavily skewed to one side. However, the true local maximum should still reside in the range $(k-1,k+1)$. To achieve a better estimate of the true location of the peak, a three-point Newton interpolation of the local discrete curve is applied. The Newtonian curve interpolation of the original continuous distribution, $h(t)$, using the three consecutive points around the peak at bin k can be expressed as is shown in Equation 15.

$$h(t) = c_0 + c_1(t-(k-1)) + c_2(t-(k-1))(t-k)$$

where,

$$c_0 = h[k-1]$$

$$c_1 = h[k] - h[k-1]$$

$$c_2 = \frac{h[k+1] - 2h[k] + h[k-1]}{2}$$

Equation 15

By taking the first derivative of the approximation curve, *h(t)*, in Equation 15 and setting this derivative equal to zero, an interpolated local maximum point can be found as is shown in Equation 16.

$$t = \frac{2k-1}{2} - \frac{h[k] - h[k-1]}{h[k+1] - 2h[k] + h[k-1]}$$

Equation 16

This interpolated value, *t*, lies in the range (*k-1*,*k+1*) and can be taken as the location of the peak near interval bin *k*. After every update of the distribution, an updated list of dominant inter-onset interval peaks is created and passed to the final stage of the beat prediction system for use as beat period hypotheses.

5.3 Beat Prediction Node Pool

The beat prediction node pool is the final element in the proposed beat prediction system and is responsible for all beat detection and prediction. It contains a collection of independent agent nodes, each representing a unique beat period hypothesis, that compete for selection as the best guess at the actual beat. Each node has a score that represents the strength of its beat prediction and the node with the highest score is selected to represent the system's guess at the actual beat. This node is then used to generate future beat predictions.

Unfortunately, selecting the node with the highest score at any given time to generate the beat prediction output does not always yield satisfactory results. It is common for two or more nodes to have equally strong beat predictions – for example,

60 BPM and 120 BPM in a song at 120 BPM – and selecting the node with the highest score can result in unwanted oscillation between these nodes. Therefore, for a node to be selected as the best or winning node, it must maintain the highest score for a time no less than one uninterrupted second. After this time period, the new best node can be selected and will not be replaced until another node maintains the highest score for at least one second. This arbitrary one-second delay acts as a rudimentary winning node decision hysteresis.

5.3.1 Agent Node Creation and Destruction

As stated earlier, the inter-onset interval statistic collector provides a continually updated list of possible beat periods. This list is used to guide the creation, modification and destruction of beat detection agent nodes. With every update to the dominant IOI list, three actions occur:

1. **Creation:** The dominant IOI list is first compared to the existing beat period hypothesis agents. A match is found if a list entry and an agent period differ by less than a given tolerance, in this case equal to 10 ms. Any unmatched list entry is used to create a new beat detection agent with period hypothesis equal to this unmatched dominant IOI.

2. **Modification:** All matched agent nodes receive the new inter-onset interval peak value and are permitted to modify their beat period hypothesis if necessary. The details of this operation are covered later in Section 5.3.2.4.

3. **Destruction:** If a node is not matched to a dominant peak in the IOI histogram and the node's score is below a minimum threshold, the node is removed. The minimum node score threshold used is zero; nodes with negative score and no matching dominant IOI are likely not to represent a valid beat hypothesis and can be removed. A node is also removed if more than one node matches a single list entry. In the case that multiple nodes match one list entry, all matching nodes except the node with the highest score are removed. This has the effect of removing duplicate beat hypotheses should they ever occur.

5.3.2 Nodes

Every component of the proposed system discussed up until this point serves as either data preprocessing or system framework. It is the beat hypothesis agent nodes that perform all of the beat detection, tracking, and prediction. Each agent node in the node pool represents a single beat hypothesis and acts mostly independently of all other nodes. A node must use the onset stream to find strong periodicities at its unique period and attempt to lock onto these patterns, track them, and predict future beat locations. A recurrent timing network forms the heart of each node and is responsible for much of the beat detection and prediction work. However, there are many more components within each node that are necessary for robust beat detection.

One of the most important features of the proposed system is period self-adjustment. When a node is created to track a given period hypothesis, it is unlikely that the period of the actual beat within the music is, and will remain, *exactly* equal to the original hypothesis. This is frequently a result of the system's inability to exactly predict the dominant inter-onset interval length or the tendency for the performing artists to change the song tempo, to a small or large degree, during play. Moreover, even if neither of these cases applied, the beat period prediction would have to be extremely accurate from the beginning to avoid the predicted beat from slowly drifting away from the actual beat. For example, a period error as little a 1 ms in a 0.5 sec period would amount to a prediction phase error of 10% after only 25 seconds. For there reasons, each beat detection node must constantly re-evaluate and slightly adjust its beat period hypothesis in order to keep the predicted and actual beats in synch. This process addresses the susceptibility of the recurrent timing networks to pattern jitter and non-integer period lengths as discussed in Section 4.1.

Consequently, every node is responsible for determining possible periodicities at its hypothesis period, finding the strongest periodicity to label as the detected beat, performing period hypothesis self-adjustment to maintain synchrony, and calculating a node score based on how likely the node is to represent the actual beat of the music. To perform each of these tasks, each node has a variable rate down-sampler, a recurrent

timing network, a beat detection and prediction logic block, a down-sample rate controller, and a node score generator. Figure 9 shows the structure and connectivity of these internal node components, each of which is discussed in detail below.

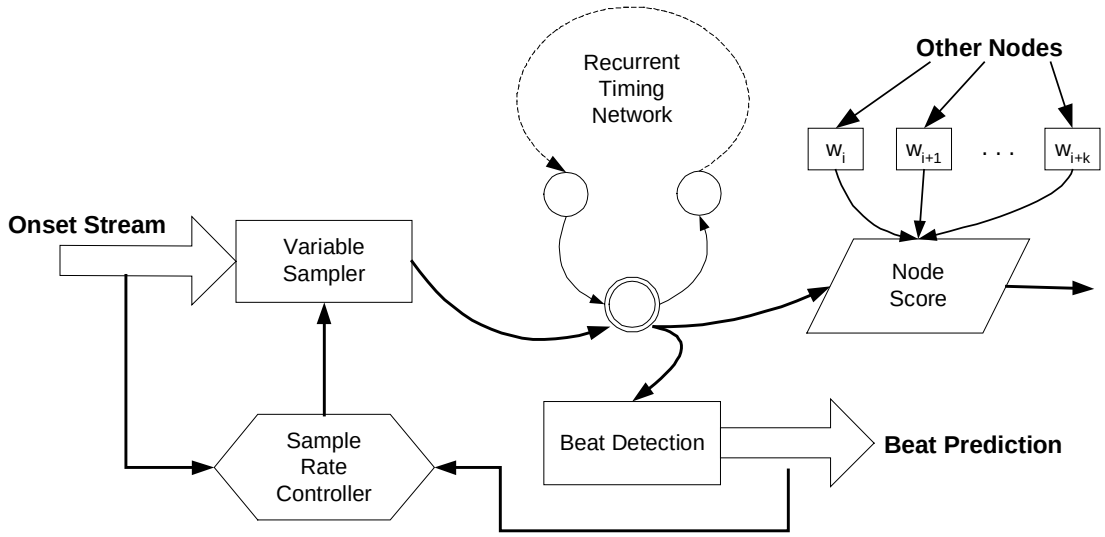

Figure 9: Beat detection agent node internal component structures

5.3.2.1 Variable Rate Down-Sampler

Every beat detection agent node receives the onset stream directly from the onset detection unit and uses this signal to attempt to find the beat. Each node must be able to finely adjust its period hypothesis while attempting to track the beat of the music. The period under consideration in a particular node is selected by setting the delay length of the node's recurrent timing network. Recall that the length of the delay in a recurrent timing network is set by altering the number of neurons in the network. Equation 17 shows the node period hypothesis, p_h, as a function of the integer number of neurons in the recurrent timing network of the node, N_p, and the sample rate of the input signal, p_{input}. Since the activation levels in the network are cycled once per incoming sample, setting the delay length of the network by changing the number of neurons, N_n, only allows the selection of periods that are integer multiples of the input sampling frequency. Regrettably, this method of period hypothesis selection is far too

coarse. Extremely fine period alteration is necessary to properly track the pulse of the song.

$$p_h = \left\| N_p \right\| \cdot p_{input}$$

Equation 17

Since the recurrent timing network must always have a delay of an integer multiple of the sample rate, the only way to finely tune a node's period hypothesis is to allow each node to slightly alter the sample rate of the incoming onset stream. Consequently, before allowing the recurrent timing network to process the onset stream, it is down-sampled by a factor of approximately nine to a target sample rate near 50 Hz. This target sample rate can be continually adjusted while leaving the length of the recurrent timing network untouched to enable the tracking of a wide range of beat periods. Through this process, the period hypothesis can be expressed as shown in Equation 18 where p_s is the new sampling period, a real multiple, K_{ds}, of the input sampling period. If the input signal is down-sampled to the sampling rate of p_s, the node's period hypothesis becomes easily adjustable using the parameter K_{ds}.

$$p_h = \left\| N_p \right\| \cdot p_s$$
$$\text{where,}$$
$$p_s = p_{input} \cdot K_{ds} \ , \ K_{ds} \in \Re$$

Equation 18

For example, if a node were to represent a 0.5 sec hypothesis, a network length of 25 neurons and a down-sample rate, p_s, of 50 Hz would be used. If, in another circumstance, a node were to represent a 0.505 sec hypothesis, a network length of 25 neurons would also be appropriate, but a down-sample rate of 49.505 Hz would be required. If a third node wished to represent a hypothesis period of 0.997 sec, a length of 50 neurons and a down-sample rate of 50.15 Hz would be needed. Therefore, by using the recurrent timing network length to roughly set the period hypothesis and the down-sample rate to finely alter it, any period hypothesis can be represented. Furthermore, when period self-adjustment occurs within the node, the down-sample

rate can be altered by slightly adjusting K_{ds} accordingly while maintaining the recurrent timing network length.

It is important to reiterate that each node acts, for the most part, independently. That is, each node has its own down-sample rate, p_s, and independently alters its unique rate using its own value for K_{ds}.

One of the disadvantages of discretely sampled data that is only compounded by down-sampling is the loss of temporal resolution. Because this unit is down-sampling a stream of impulses, the position of an impulse within a set of samples to be combined is lost under standard down-sampling practices. Figure 10 illustrates this problem. It is understood that the period of the beat within the music signal being processed will virtually never be an exact multiple of the sampling period. Consequently, the blind discretization process of down-sampling will only further exacerbate the occurrences of pattern jitter.

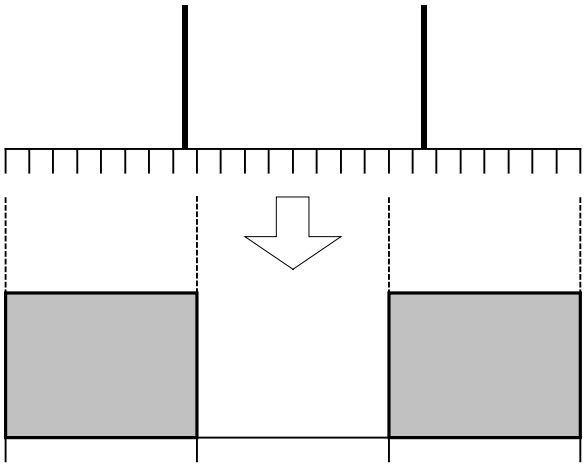

Figure 10: An extreme example of lost onset location accuracy through down-sampling

From previous discussions in Sections 5.1 and 5.2.1, it is known that the temporal location of each onset is not assumed to be entirely accurate. By replacing the impulse form of each onset with a function such as a Gaussian distribution, a crude representation of onset location probability can be made. That is, instead of stating with certainty that an onset exists at a given time, the location of each onset, at the risk

of misusing the term, is fuzzified. This way, an idea of approximate or likely onset location is born. If onsets are fuzzified before down-sampling, the energy of an onset can be split between two 50 Hz samples proportionally to its mean location. Figure 11 illustrates this behaviour. Equation 19 shows the formula used to calculate the down-sampled input signal $x[n]$ (sampling rate p_s), from the input signal $x_{input}[n]$ (sampling rate p_{input}), where $W(\theta)$ is the fuzzy onset window function.

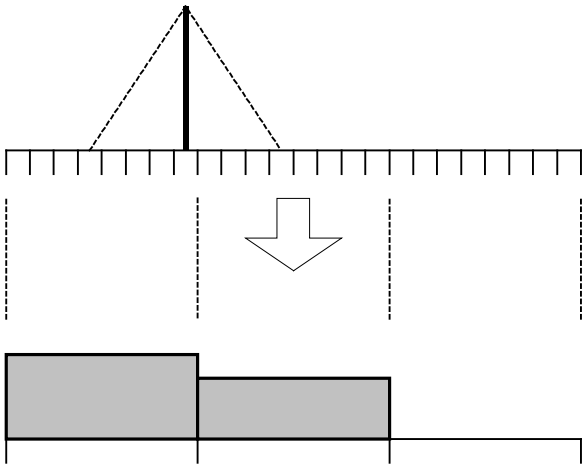

Figure 11: Spreading the probability of an onset location over a larger window before down-sampling.

$$x[n] = \sum_{i=n\cdot K_{ds}}^{(n+1)\cdot K_{ds}} \left[x_{input}[i] \cdot \int_{n\cdot K_{ds}}^{(n+1)\cdot K_{ds}} W(\theta - i)\, d\theta \right]$$

Equation 19

Note that the occurrence of an onset near the boundary of two samples gives rise to energy in both samples. This process may appear to further obscure the true location of an onset, but in actuality, it enables the recurrent timing network to see a better indication of probable onset location. This approach is similar in theory to the Parzen windowing seen in Section 5.2.1.

The selection of the new onset distribution function was somewhat arbitrary. The Normal distribution was not used due to obvious difficulties in integration and therefore a triangle distribution was selected. The width of the base of the triangle was chosen to be a conservative 20 ms. This addition of a non-impulse function onset distribution requires the system to look ahead in the onset stream by one-half of the width of the function. As a result, a delay buffer of 10 ms is needed in this unit, bringing the total system lag to 75.84 ms.

This fuzzification step occurs in the down-sampler unit of each node and not in the onset detection onset stream generation for two reasons. The first and most pragmatic reason is that impulses in the onset stream allow for easier measurement of inter-onset interval distributions as described in Section 5.2.1. The second reason involves the requirement of impulse representations of onsets for the self-adjustment behaviour of each node that will be discussed in more detail later.

The down-sampled output of this component, $x[n]$, is sent to the recurrent timing network within the node such that periodicities in the signal can be detected. The operation of the recurrent timing network in the context of a beat hypothesis agent node is explored next.

5.3.2.2 The Recurrent Timing Network

The recurrent timing network in each node is constructed to provide a delay length, in combination with the down-sample rate, exactly equal to the node's beat period hypothesis. The network is thus able to detect strong periodic elements specific to the frequency prescribed by the node. Since the operation of recurrent timing networks and the modifications made to Cariani's simple networks were already discussed in Chapter 4, they will not be covered again here.

The modified recurrent timing network of Section 4.2 requires an input in the range of [-0.5,1] to properly cause growth and decay in the network – negative inputs cause decay and positive inputs cause growth. The down-sampled onset stream contains zeros where no onsets are located and nonzero values at the locations of

onsets. The translation between onset stream levels and recurrent timing network inputs is straightforward and is given in Equation 20. Figure 12 graphically shows this translation.

$$I[n] = \begin{cases} \left((0.05 - x[n])/0.05\right) \cdot -0.5, & if \quad x[n] \le 0.05 \\ \left(x[n] - 0.05\right)/0.95, & if \quad x[n] > 0.05 \end{cases}$$

Equation 20

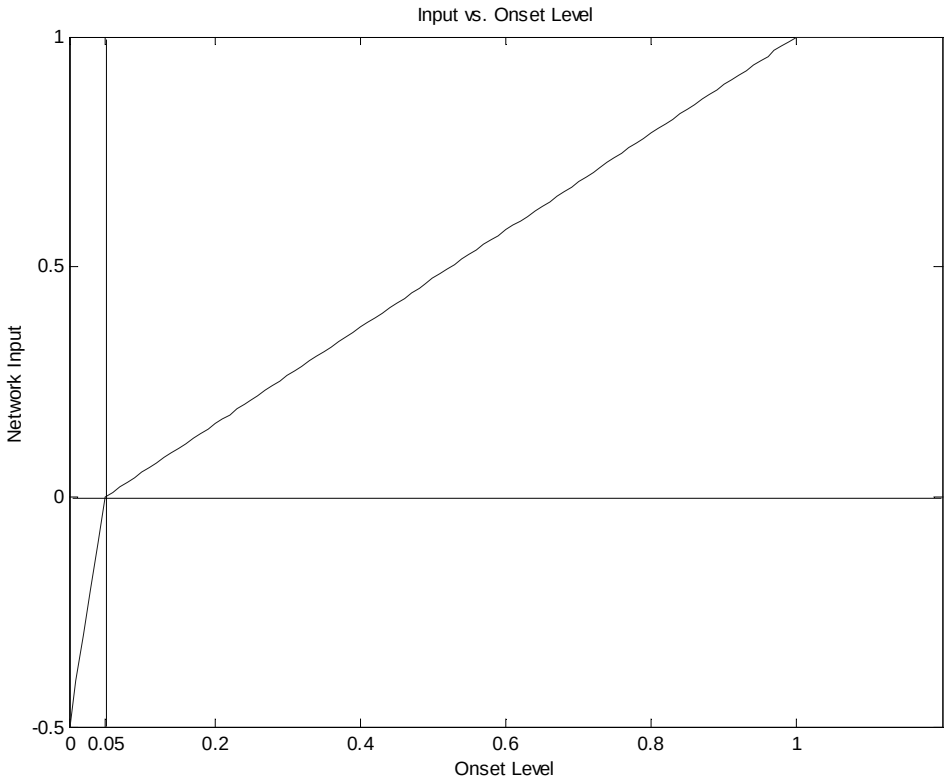

Figure 12: Onset level vs. network input level

The rationale behind this function is that an extremely weak onset is only marginally better than no onset at all. The absence of an onset and the presence of a very weak onset cause decay in the network. Onsets stronger than 5% (0.05) cause activation level growth in the network proportional to the strength of the onset.

5.3.2.3 Beat Detection and Prediction

Following the recurrent timing network in the data flow within each node is a beat detection and prediction unit. The name of this component may be slightly deceiving since the actual beat detection and prediction is, in fact, done in the recurrent timing network. This next component, however, is responsible for analyzing the contents of the recurrent timing network and determining the phase of the best or strongest detected beat. This simply amounts to finding the strongest activation level propagating around the network loop. A beat in synch with the input data to the network is predicted when this maximal activation level reaches the zero-phase, head node position. When a beat is predicted, this component generates an impulse on the output beat prediction stream. This process can be simplistically represented as in Equation 21, where $bp_{output}[n]$ is the beat prediction output, a_i is the activation of the i^{th} neuron in the recurrent timing network and a_0 is the activation level of the head node. Of course, the beat prediction output of any one node may or may not be used as the beat prediction output for the entire system.

$$bp_{output}[n] = \begin{cases} a_0, & \text{if } a_0 > a_i, \forall i \\ 0, & \text{otherwise} \end{cases}$$

Equation 21

To ensure stability, a neuron activation level must be the maximum level in the loop for two beats in a row before it can be considered the "detected beat".

At this point in the data flow, the model suffers from a total systematic delay of 75.84 ms. In order to synchronize the beat prediction output with the original music input, this delay must be reversed. This reversal is easily accomplished by generating a predicted beat output before the maximum activation propagates to the head node. By generating the beat prediction output impulse at a neuron before the head node, a delay of approximately 20 ms is cancelled per reversed neuron. Furthermore, since the recurrent timing network cycles at a sample rate of approximately 50 Hz, adjusting the exact moment within this sample period at which a beat prediction impulse is generated

can further fine-tune this delay reversal. However, due to natural human perception limitations and the many inherent inaccuracies already existing in the system, this second delay reversal step is unnecessary. Therefore, predicting a beat output four nodes before the head node, about 80 ms, is sufficient. As a result, the beat prediction output, $bp_{output}[n]$, can be expressed as seen in Equation 22, where a_{-4} is the activation level of the node four nodes before the head node.

$$bp_{output}[n] = \begin{cases} a_{-4}, & \text{if } a_{-4} > a_i, \forall i \\ 0, & \text{otherwise} \end{cases}$$

Equation 22

5.3.2.4 Down-Sample Rate Controller

Outside of the recurrent timing network, the down-sample rate controller and the node score generator are the most important components of the entire system. The down-sample rate controller is responsible for determining the amount of adjustment needed to the node's period hypothesis. By properly adjusting the node period hypothesis, small period errors, fluctuating timing, and changing tempi can be accounted for.

In this and any beat prediction or detection system, there are two possible representations of the beat: the predicted or detected beat generated by the system itself, and the actual beat of the music under consideration. The predicted and actual beat representations frequently differ and it is the goal of the prediction system to minimize this difference. Unfortunately, this is one of the most difficult challenges facing any beat detection system because the system cannot know where the actual beats lie – if a system did know, the detection problem would, of course, be trivial. Herein lies the circular problem of node period hypothesis adjustment.

The tempo or period hypothesis of a node should be changed and updated such that the generated beat prediction continues to coincide as closely as possible with the location of the actual beat. Since the actual beat location is unknown, a best guess must be made such that a node may change its period appropriately. The assumption that

many onsets must lie on the beat can be applied, once again, in this case to create an actual beat location estimate. The goal of this period modification component can then be rephrased to ensure that the generated beat prediction should frequently coincide with the location of an onset – an onset hopefully coinciding with an actual beat.

For example, if an onset were to arrive a few milliseconds before a predicted beat, there is a chance that the predicted beat period is slightly too long. If, on the next cycle, an onset were to appear again a few more milliseconds before the predicted beat, it is likely that the predicted period is incorrect – it is too long. The sample rate controller, in this case, should reduce the beat period hypothesis for the current node to try to realign the predicted beats with the occurrence of input onsets.

More formally, the goal of the sample rate controller is to adjust the down-sample rate in order to reduce the prediction error, the error detected between the predicted and actual beat, to zero. There are two forms of beat prediction error that can occur while processing a section of music with unchanging tempo. The first error form is a phase error, the absolute error between a predicted beat and the actual beat. If the predicted beat period is precisely equal to the actual beat period, the phase error will remain constant.

The second form of prediction error is a period error, the difference between the predicted and the actual period of the beat. This error is manifested in the change in phase error between two successive beat predictions. An increasing phase error with constant slope indicates a period error equal to that slope. In other words, period error is equal to the derivative of phase error and the occurrence of period error inevitably results in a phase error. Equation 23 shows an expression for the true beat prediction error, $e[n]$, expressed as a function of the actual beat location, $b_a[n]$, and the hypothetical or predicted beat location $b_h[n]$. This error can also be expressed recursively, as a function of the phase error, $e_\phi[n]$, and the period error, $e_\rho[n]$. Of course the phase error is simply equal to the prediction error and the period error is a function of the actual and predicted beat periods, $p_a[n]$ and $p_h[n]$ respectively or the first difference in phase or prediction errors.

$$
\begin{aligned}
e[n] &= b_h[n] - b_a[n] \\
&= e_\phi[n-1] + e_\rho[n-1] \\
e_\phi[n] &= e[n] \\
e_\rho[n] &= \left(p_h[n-1] - p_a[n-1]\right) = e[n] - e[n-1]
\end{aligned}
$$

Equation 23

Since these actual errors cannot be known, estimates using onset locations must be made. The formula for the prediction error estimate can be found in Equation 24 where $o[n]$ gives the time of the closest onset to the n^{th} beat. The ^ mark above a symbol indicates an estimated value.

$$
\begin{aligned}
\hat{e}[n] &= b_h[n] - \hat{b}_a[n] \\
&= b_h[n] - o[n]
\end{aligned}
$$

Equation 24

In order to correct both phase and period errors, the node's period hypothesis must be altered. To approach this problem, ideas from the design of a proportional differential gain controller from basic control systems theory [31] were applied. Equation 25 gives a control theory-based function for updating the node period hypothesis given the measured or estimated prediction error. The proportional gain is given by K_p and the derivative gain is given by K_d.

$$
p_h[n] = p_h[n-1] - K_p \cdot \hat{e}[n] - K_d \cdot \left(\hat{e}[n] - \hat{e}[n-1]\right)
$$

Equation 25

Substituting Equation 25 into the formula for period error in Equation 23, an expression for the estimated period error shown in Equation 26 can be found.

$$
\begin{aligned}
\hat{e}_\rho[n+1] &= \hat{e}[n+1] - \hat{e}[n] \\
&= p_h[n] - \hat{p}_a[n] \\
&= \left[p_h[n-1] - K_p \cdot \hat{e}[n] - K_d \cdot \left(\hat{e}[n] - \hat{e}[n-1]\right)\right] - \hat{p}_a[n]
\end{aligned}
$$

Equation 26

The assumption was made earlier that the actual tempo of the song is unchanging. For this reason, $p_a[n]$ is constant for all n. The difference between estimated actual period and hypothetical period in Equation 26 can then be substituted given the identity in Equation 23. Equation 27 shows the result of this substitution.

$$\begin{aligned} p_h[n-1] - \hat{p}_a[n] &= p_h[n-1] - \hat{p}_a[n-1] \\ &= \hat{e}[n] - \hat{e}[n-1] \end{aligned}$$

$$\begin{aligned} \text{so then, } \hat{e}_\rho[n+1] &= \hat{e}[n] - \hat{e}[n-1] - K_p \cdot \hat{e}[n] - K_d \cdot \left(\hat{e}[n] - \hat{e}[n-1]\right) \\ &= \left(1 - K_p - K_d\right) \cdot \hat{e}[n] + \left(K_d - 1\right) \cdot \hat{e}[n-1] \end{aligned}$$

Equation 27

If this expression is then substituted into the recursive error definition from Equation 23, Equation 28 arises.

$$\begin{aligned} \hat{e}[n+1] &= \hat{e}[n] + \hat{e}_\rho[n] \\ &= \hat{e}[n] + \left(1 - K_p - K_d\right) \cdot \hat{e}[n] + \left(K_d - 1\right) \cdot \hat{e}[n-1] \\ &= \left(2 - K_p - K_d\right) \cdot \hat{e}[n] + \left(K_d - 1\right) \cdot \hat{e}[n-1] \end{aligned}$$

Equation 28

Examining Equation 28 reveals a trivial solution to this control problem. By setting both proportional gain, K_p, and differential gain, K_d, equal to one, the estimated error can always be reduced to zero. Equation 29 gives the formula for updating the node hypothesis period given the estimated prediction error. Of course this only applies to the ideal steady-state (zero tempo change), noise-free case when the estimated error is a good representation of the actual prediction error.

$$\begin{aligned} p_h[n] &= p_h[n-1] - 1 \cdot \hat{e}[n] - 1 \cdot \left(\hat{e}[n] - \hat{e}[n-1]\right) \\ &= p_h[n-1] - 2\hat{e}[n] + \hat{e}[n-1] \end{aligned}$$

Equation 29

Removing the restriction of zero tempo change does not adversely affect the performance of this control function. Although the function does not account for the

fact that the actual beat period could be changing, it can be assumed that any tempo change occurs sporadically. In other words, the tempo is assumed not to be steadily increasing or decreasing over an extended period of time. Such behaviour could be accounted for in the controller by adding a tempo change error equal to the second difference of the prediction error, but this is beyond the scope of this research.

Unfortunately, assuming that the prediction error estimation is accurate is not valid for actual application in the sampling rate controller. Recall that the error is estimated by assuming that selected onset locations are equal to the actual beat locations. Onsets do not coincide with a beat on every cycle, and when they do, they often suffer from significant error or expressive timing. By applying the node period update function shown in Equation 29, it was found that significant over-fitting occurred. Specifically, one spurious onset could generate a large error estimate that results in wild period fluctuations. Changes need to be made to the period hypothesis update procedure to correct this problem.

The validity of the error estimation used to update the period in Equation 29 hinges on the assumption that onsets coincide exactly with the actual beat. In fact, most of the theory in the proposed beat detection system assumes this same fact. What is important to note is that where the prediction error estimate uses the location of a single onset, the proposed system uses trends of onset locations to predict the actual beat.

If error is introduced into the error estimate, $\hat{e}[n]$, this error is transfer directly to the node period. To combat this, it is necessary to measure the likelihood that a given error estimate, $\hat{e}[n]$, is close to the true prediction error, $e[n]$. If the estimated or measured error is unlikely to be representative of the true error, the influence of Equation 29 should be minimized. Conversely, if the likelihood is high, Equation 29 should be applied as it yields accurate results.

The determination of error measurement validity is not trivial. Once again, the true validity cannot be known and must itself, be estimated. One possible solution could involve smoothing or averaging the onset locations over a short sequence of

measurements to get a better idea of the trend of the actual beat location. If the measured onsets are distributed around the actual beat location with zero mean – a reasonable assumption – an average of onset locations should yield a good actual beat location estimate. The problem with this approach is that the averaging process causes significant lag in measurement error that results in a slow period correction response. This usually causes non-decreasing error oscillation and significant loss of activation energy within the recurrent timing network.

A slightly more enlightened approach was inspired by the work of Jones and Large in [1]. They suggest the use of "expectation windows" that surround each predicted beat location. When an onset is detected in the vicinity of a predicted beat, the expectation level gives an indication of the level of expectancy attributed to the onset – that is, how expected an onset at this location is to the system. Jones and Large use this expectation window to weight the effect the presence of an onset has on their period and phase predictions of a periodic event. A similar approach can be used here.

In each node with a somewhat valid beat period hypothesis, large activation levels appear in its recurrent timing network. The largest activation in each network can be assumed to represent a strong detected beat likely coincident with the actual beat of the song. The detected beat can then be used as the node's beat prediction. In fact, the detected beat in the recurrent timing network is arguably the best estimate of the actual beat location. This is logical since the goal of the entire system is to predict the actual beat location with as much accuracy as is possible. Consequently, onsets that are near to the existing beat prediction are most likely to coincide closely with the actual beat assuming the actual beat has changed only slightly over the previous few cycles. As depicted in Figure 13, an expectation window that surrounds the detected beat location can be used to weight each onset with a value indicative of its likelihood for coinciding with an actual beat. This will work in all cases where the actual beat does not change from the thus-far detected beat with great speed. This process may once again seem circular, and it is, but it is effective.

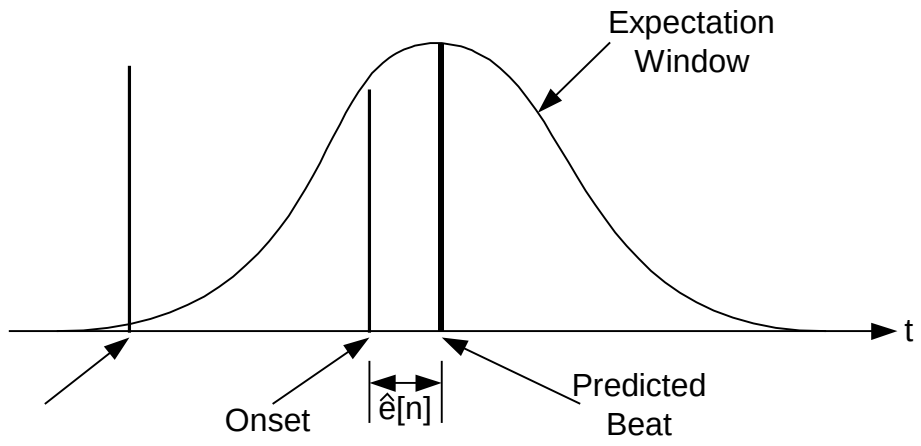

Figure 13: A Gaussian expectation window surrounding a predicted beat used to weight prediction error validity

The use of an expectation window that surrounds the predicted or detected beat to weight the validity of onset measurements also solves another minor problem. Recall that occasionally, no onset may coincide with a given beat. When this happens, the prediction error estimate, as defined above, is impossible to calculate. Modification of the beat period hypothesis should then be deferred until the next cycle in which a valid onset can be found. Using an expectation window similar to the one shown in Figure 13 solves this problem implicitly by assigning a near-zero weight to onsets that are too far from the predicted beat. Therefore, only onsets that are likely to coincide with an actual beat are given a significant weight.

The addition of an expectancy weighting term into the period update function is fairly trivial. Equation 30 shows the method for calculation of the expectancy weighting for predicted beat, $b_h[n]$, using the closest onset, $o[n]$. A Gaussian windowing function with standard deviation σ_E is used. Krumhansl [32] states that listeners are sensitive to approximately an 8% time difference between successive periodic events. This value is used to set the standard deviation of the expectancy window: 8% of the period of the node.

$$w_E[n] = \exp\left\{-\left(b_h[n] - o[n]\right)^2 / \sigma_E{}^2\right\}$$

Equation 30

Equation 31 gives the new update function with the appropriate weights. Since two prediction error estimates are used in calculating the new node period, two onset expectancies must be combined using a geometric average to determine an appropriate weighting for Equation 29.

$$p_h[n] = p_h[n-1] - w_E[n] \cdot \hat{e}[n] - \sqrt{w_E[n] \cdot w_E[n-1]} \cdot \left(\hat{e}[n] - \hat{e}[n-1]\right)$$

Equation 31

It was mentioned in Section 5.3.1 that even after creation, the IOI histogram predicted beat period helps steer the node's period. The updated node period is calculated as specified above unless the expectation weighting becomes low, in which case the IOI histogram predicted period is allowed to draw the period toward a "safe" value. Essentially, when reasonable prediction error estimates are made (the expectancy of onsets is high), the IOI predicted period is essentially ignored. However, if the arriving onsets start to drift significantly from the predicted beat, the estimated actual beat seems no longer to be valid and the IOI dictated beat estimate is given more weight in hopes that it may help bring the node back to a more representative period hypothesis.

Equation 32 shows this final version of the period update function where $p_{PD}[n]$ is the controller modified period from Equation 31, $p_{IOI}[n]$ is the IOI dictated period, and $w_E[n]$ is the expectancy window weighting results. Note that when the expectancy weights are low, $p_{PD}[n]$ is approximately equal to $p[n-1]$.

$$p[n] = p_{PD}[n] + 0.5 \cdot \left(1 - \sqrt{w_E[n] \cdot w_E[n-1]}\right)\left(p_{IOI}[n] - p_{PD}[n]\right)$$

Equation 32

The updated period from the above equation is applied to the node by modifying the down-sample rate according to Equation 33. In this function, $p_s[n]$ is the down-sampling period, and $||N||$ is the length of the recurrent timing network within the node.

$$p_s[n] = p[n] \Big/ \|N\|$$

Equation 33

5.3.2.5 Node Score Calculator

The final node component is responsible for calculating the node score used by the system to select the node that best represents the actual beat in the musical input. This node score is intended to represent the strength of the beat prediction made by the node and lies in the range of [-1,1]. A node's score is initialized to zero upon its creation and is increased or decreased according to the validity of its beat prediction thereafter. After each sample of the onset stream is processed, the score is updated according to Equation 34 where $Score_p[n]$ is the score of the node with period hypothesis p at time-step (sample) n. The score is changed by $\Delta Score_p[n]$, which will be discussed in detail, and a scaling factor of β, calculated as shown. This scaling factor has the effect of bounding the node score between its minimum and maximum values.

$$Score_p[n] = Score_p[n-1] + \beta \cdot \Delta Score_p[n]$$

$$\beta = \begin{cases} 1 - Score_p[n-1], & \text{if } \Delta Score_p[n] \geq 0 \\ Score_p[n-1] + 1, & \text{if } \Delta Score_p[n] < 0 \end{cases}$$

Equation 34

The score change value, $\Delta Score_p[n]$, is calculated using two important factors. First, the data within the timing network is used to calculate an energy value for the node. Second, information from other nodes is used to reinforce the scores of those nodes whose predictions are the most consistent between all nodes in the system.

The energy contained within the recurrent timing network provides a good indication of the strength of a node's beat prediction. Nodes with defensible beat period hypotheses will have large concentrations of energy in their recurrent timing networks at the locations of periodic elements such as beats or off-beats. Nodes

representing invalid or weak beat hypotheses will fail to find input stimuli periodic at their period hypothesis, and as a result, their recurrent timing networks will contain only low activation levels.

Node strength or network energy is calculated using a simple weighted root-mean square of the activation levels in the recurrent timing network as is shown in Equation 35. In this formula, E_p is the energy or strength of the network in the node with hypothesis, p, $||N_p||$ is the length, in neurons, of the recurrent timing network, and a_i is the activation level of the i^{th} neuron within the network. The RMS of the activation levels is weighted by the square root of the length of the network such that if two networks have equivalent energy content, the shorter network is preferred. The reason for this is that since only one beat is output per loop duration, shorter loops are preferred since beats are generated more frequently. This minimizes the chance that a large activation spike is located within the network of the winning node but is not used to generate a beat output.

$$E_p = \frac{1}{\sqrt{\left\|N_p\right\|}} \cdot \sqrt{\frac{\sum_{i \in N_p} a_i^{\,2}}{\left\|N_p\right\|}} = \frac{\sqrt{\sum_{i \in N_p} a_i^{\,2}}}{\left\|N_p\right\|}$$

Equation 35

This node energy value is low-pass filtered in time to smooth the behaviour of this instantaneous network strength measurement. This ensures that occasional spikes in the network energy do not propagate through to the node score itself. Equation 36 shows this filtering process where α is the decay parameter, selected to equal 0.9.

$$\overline{E}_p = \alpha \cdot \overline{E}_p + (1-\alpha) \cdot E_p$$

Equation 36

While the strength of the predicted beat within each node gives a good indication of the likelihood that the node represents the actual beat within the song, it is not always sufficient. It is possible for a node to contain ample energy in its recurrent

timing network even though it does not predict the actual beat of the input music. For this reason, an important part of the proposed system is allowing information from all node period predictions to aid in the discovery of the true beat.

The allowed range of node period hypotheses in the proposed system is quite wide, and hence, it is likely that a few nodes represent different levels of the input music's metrical hierarchy. For example, music with a true beat period of 0.5 seconds also may have metrical levels at periods of 0.25 and 1.0 seconds. All three of these hypothetical metrical levels should be represented by three different nodes in the system. In theory, a node that is representative of the true beat should be accompanied by strong nodes on other levels of the metrical hierarchy. According to Lerdahl and Jackendoff [5], these other levels should reside at integral harmonic and subharmonic frequencies. A robust beat detection or prediction system should use the existence of strong accompanying metrical levels to strengthen the appropriate prediction. Large used this realization in his coupling between oscillators in his 2000 paper [2].

In the system described here, a node's score is increased by the presence of strong affirming nodes and is decreased by the presence of strong conflicting nodes. This is done by calculating the node score change value as a weighted sum of the network energies of all nodes in the system. This process is shown in Equation 37 where $w_{p,i}$ is the weight used to calculate the contribution of node i to the score of node p and $\bar{E}_i[n]$ is node i's network energy as calculated in Equation 36. The weight within the sum given to the current node's network energy, $w_{p,p}$, is a constant set to be much larger than the maximum weight from external nodes.

$$\Delta Score_p[n] = \sum_{i \in System} w_{p,i} \cdot \overline{E}_i[n]$$

Equation 37

The inter-node weights, $w_{p,i}$, are calculated to best correspond to the likelihood that the two nodes in question represent an equivalent beat hypothesis and lie within the range of [-0.03, 0.04]. Positive weights are given to nodes with supporting period hypotheses and negative weights are given to nodes with conflicting hypotheses.

While it is reasonable to use the concepts of metrical levels to dictate the coupling strength between nodes, as was done by Large [2], a different approach is taken here. In the proposed system, weights between nodes should correspond to the degree to which the two nodes represent an equivalent beat hypothesis. If the beats predicted by one node always coincide with those of a second node, this second node is considered in full support of the first node. Nodes in full support should be given the maximum weight. Larger infrequency of coincidence results in less support and, consequently, a lower inter-node weight.

Instead of explicitly determining the degree of coincidence between two beat predictions, a theoretical calculation is made, based on beat periods alone, to guide the weights. The work of Scheirer [4] shows that a comb filter with a specific period can be used to determine the extent of a signal's periodic agreement with the comb filter's period. The comb filter acts as an ideal beat stream where the beats occur as impulses. When this beat stream is multiplied and summed (i.e. convolved) with a signal at a given phase, the degree of correlation between the two signals can be measured. If the signal under consideration were to be an ideal stream of beats represented by impulses, the result of convolution with a comb filter would indicate the degree of coincidence of the beats of the two signals – that is, the coincidence of the beats represented by the comb filter and the beats represented by the impulse train. In this manner, the comb filter technique can be used to determine the coincidence between two beat predictions. The comb filter represents the current node's beat hypothesis, and the impulse train represents the beat prediction of a node in question.

Mathematically, the formulation of the two signals, comb filter and beat impulse train, can be expressed as in Equation 38 where $h[n]$ is the comb filter signal and $x[n]$ is the beat stream under consideration. In this equation, p_B is the period of the beat stream created from a node B, p_A is the period of the comb filter created from a node A, and α is the comb filter decay parameter.

$$x[n] = \sum_{\forall k \in I} \delta[n - k \cdot p_B]$$
$$h[n] = \sum_{k \geq 0} \alpha^{k+1} \cdot \delta[n - k \cdot p_A]$$

Equation 38

Assuming the two periods, p_A and p_B are integers, the convolution of the two signals will yield an impulse train periodic in p_B, containing L different impulses per period. These resulting impulses correspond to coincidence between some elements of the comb filter and elements of the beat train. Since only the degree and not the phase of coincidence is important, the magnitude of these L possible unique non-zero values from the convolved signal can be expressed as shown in Equation 39. In this equation, $y_{A,B}[l]$ is a signal of length $L_{A,B}$ containing the unique magnitudes resulting from the convolution $x[n]*h[n]$. The term, $L_{A,B}$ corresponds to the number of unique non-zero values resulting from this convolution.

$$y_{A,B}[l] = \sum_{k=0}^{\infty} \alpha^{\left[l + k \cdot L_{A,B}\right]}, \; l \in [1, L_{A,B}]$$
$$= \frac{\alpha^l}{1 - \alpha^{L_{A,B}}}$$

Equation 39

The value of $L_{A,B}$ is equal to the number of time-steps or samples that $x[n]$ and $h[n]$ coincide in one period of $x[n]$. It is also equal to the integer number of beat periods of node *A* needed to align with an integer number of beats periods in node *B*. This value is proportional to the lowest common multiple of the two beat periods, p_A and p_B, and can be found using Equation 40. Since it is possible, or in fact, likely, for p_A and p_B to be non-integers, the LCM function must be calculated approximately. This is done by finding integer multiples of each of p_A and p_B that differ by less than some small tolerance, ε.

$$L_{A,B} = \frac{\mathrm{LCM}(p_A, p_B)}{p_A}$$

Equation 40

The expression in Equation 39 gives the $L_{A,B}$ different coincidence values between the two beat hypotheses *A* and *B*. However, to get a representative number indicating the total degree of concurrence of the hypothesis *B* with the node *A*, an average of these values is required. This average can be expressed as is shown in Equation 41 where $C_{A,B}$ is the concurrence value of node *B* with node *A*.

$$C_{A,B} = \frac{\sum_{l=1}^{L_{A,B}} y_{A,B}[l]}{L_{A,B}}$$

Equation 41

Given Equation 39, the numerator of Equation 41 can be expanded as is shown below in Equation 42.

$$\sum_{l=1}^{L_{A,B}} y_{A,B}[l] = \sum_{l=1}^{L_{A,B}} \sum_{k=0}^{\infty} \alpha^{[l+k\cdot L_{A,B}]}$$

$$= \sum_{l=1}^{L_{A,B}} \left[\alpha^{l} + \alpha^{l+L_{A,B}} + \alpha^{l+2L_{A,B}} + \alpha^{l+3L_{A,B}} + \ldots\right]$$

$$= \alpha^{1} + \alpha^{1+L_{A,B}} + \alpha^{1+2L_{A,B}} + \ldots$$

$$\ldots + \alpha^{2} + \alpha^{2+L_{A,B}} + \alpha^{2+2L_{A,B}} + \ldots$$

$$\ldots + \alpha^{3} + \alpha^{3+L_{A,B}} + \alpha^{3+2L_{A,B}} + \ldots$$

$$\ldots$$

$$\ldots + \alpha^{L_{A,B}} + \alpha^{2L_{A,B}} + \alpha^{3L_{A,B}} + \ldots$$

$$= \alpha^{1} + \alpha^{2} + \alpha^{3} + \alpha^{4} + \alpha^{5} + \ldots$$

$$= \sum_{i=1}^{\infty} \alpha^{i}$$

$$= K \quad \Theta\ 0 < \alpha < 1$$

Equation 42

Recall that α is the comb filter decay parameter and so it must lie in the range (0,1). Consequently, Equation 42 reduces to a constant, *K*, and as such Equation 41 is found only to depend on $L_{A,B}$. Since this constant, *K*, is a common scaling factor to all concurrence values, $C_{A,B}$, it can simply be divided out to unity (equivalent to setting α equal to a reasonable decay of 0.5). As such, the average concurrence value of node *B* with node *A* calculated Equation 41 simply becomes equal to one over $L_{A,B}$. In conclusion, the weight used for the score of node *A* as a result of the energy value in node *B* can be calculated using this concurrence value as shown in Equation 43. In this formula, W_{min} and W_{max} are the minimum and maximum weights respectively. As stated earlier, these limits are –0.03 and 0.04. These weights are used in Equation 37 to calculate the node score change value.

$$w_{A,B} = W_{\min} + (W_{\max} - W_{\min}) \cdot C_{A,B}$$
$$= W_{\min} + (W_{\max} - W_{\min}) \cdot \frac{1}{L_{A,B}}$$

Equation 43

Table 2 shows some sample node periods and the weights between them as calculated using Equation 43.

Table 2: Example weights between nodes

Node *A* Period	Node *B* Period	$w_{A,B}$	$w_{B,A}$
1.0 s	0.5 s	0.0400	0.0050
1.0 s	0.25 s	0.0400	-0.0125
1.0 s	2.0 s	0.0050	0.0400
1.0 s	0.66 s	0.0050	-0.0067
1.0 s	0.9 s	-0.0222	-0.0230
1.0 s	0.2 s	0.0400	-0.0160

5.4 Implementation Details

This section is meant to briefly outline some implementation details. Because the proposed model is implemented on a digital computational device, all input and output data are discrete-time sampled digital signals. The input audio data is formatted as a one-channel PCM audio stream at a sample rate of 44100 samples per second with a bit depth of 16 bits per sample. Down-sampling of the data within the system results in an output signal at approximately 50 samples per second containing non-zero values where beats are predicted. Beats are said to be predicted at the centre of these non-zero output values.

The system was implemented using C++. While the model is designed to be able to be implemented for real-time processing, the implementation created for this paper for design and testing purposes does not operate in real-time. Instead, it was implemented to process the data causally and remains real-time capable.

Chapter 6: Results and Analysis

The analysis and objective comparison of beat detection systems remains a contested issue. That the purpose of the system dictates the design of a performance measure is one of the major difficulties. A system that attempts to model human behaviour should react in a manner as similar to a human subject as possible whereas a system designed to analyze expressive timing deviations in performances must maintain a very high degree of beat location accuracy with less importance placed on human-like behaviour. In some applications, the system's output may be later scrutinized and so spurious detected beats could easily be removed. In other applications, the output may be used in real-time and therefore spurious detected beats may cause significant problems.

The beat prediction model proposed here is intended for use in real-time synchronization applications and therefore, detection of a representative, steady, and reasonably accurate beat is important. In order to determine whether a detected beat is representative of the actual beat in the music, a source of true beat locations is needed. Since the beat is a human perception phenomenon, it is reasonable to compare a beat detection system's response with the results of a human subject. However, this can pose a variety of problems.

The first problem with comparing system results to human beat detection is that there is no single true definition of the beat in a piece of music to which all listeners tap their foot or clap their hands. The beat, or tactus, is simply the pulse of one of the metrical levels in the music found to be most comfortable for a given listener to follow. Consequently, one listener may choose to hear the beat twice as frequently as another listener. In a particular song, the "true" tempo may be 60 BPM, 120 BPM, or 240 BPM. All perceptions are equally valid, so which one is the true beat to which the proposed system's output should be compared?

The second problem is that by no means is the human mind a flawless beat detector. In fact, in cursory listening tests conducted by Scheirer on a small set of experienced musicians, a root-mean-squared deviation of 80 ms or more between subjects' beat timings were not uncommon [4]. Human subjects, even experienced musicians, cannot be relied upon to generate a consistent and accurate representation of the beat.

These two issues aside, the human listener still remains one of the most reliable beat detection systems in existence as nearly all beat detection systems perform in a substandard fashion with respect to human performance. Moreover, music itself is likely to contain many timing inaccuracies and deviations such that the definition of the "true" beat is further obscured. Once again, since the beat is a human perceptual phenomenon, it makes the most sense to compare a system's results with that of a human subject.

In the analysis of the system described in the previous chapters, beat prediction results are compared and contrasted with "expert" human beat detection results. These "expert" or "true" beat locations were created by the author using careful scrutiny, both visual and acoustic, with the aid of a variety of software packages. The most notable software package used was Sonic Foundry's ACID Pro *BeatMapper* wizard [33], a package that allows a user to interactively modify and confirm possible detected beat locations. Each song was examined many times and beat locations were adjusted in an attempt to achieve the most accurate representation of the beat for each song under consideration. These perceived beat locations are used as the ground-truth "expert" beat detection results against which the output of the proposed system is compared.

Beyond comparing the proposed system's results to the "expert" beat detection ability of a human subject, it is instructive to contrast system performance with that of another successful computational beat detection model. The system proposed by Scheirer in [4] provides an excellent basis for results comparison, using beat prediction output generated using Scheirer's implementation of his model [34].

6.1 Performance Measures

Not surprisingly, there is no universally accepted method for comparison of a beat detection system's results to ground-truth beat locations or to the results of another beat detection system. Some authors use subjective evaluations of beat output correctness [4] while others attempt to apply some quantitative measure [14, 35]. The next three subsections will briefly describe two quantitative measures followed by a discussion of the approach used to evaluate the proposed model.

6.1.1 Cemgil et al's Performance Measure

Cemgil, Kappen, Desain and Honing [14] propose a simple quantitative total measure for a beat detection system meant to roughly approximate the percentage of correctly detected beats. The measure finds the closest predicted beat to each actual beat and applies a degree of match function to this error distance. A distance of zero is given a maximum weight of unity and large distances receive weights near zero. These degrees of match are averaged across the mean number of actual and predicted beats giving a good indication of the total degree of match. Through this method, Cemgil et al's measure, herein simply called the Cemgil measure, yields a 100% match only when the predicted beats have zero-deviation and one to one correspondence with the actual beats.

Equation 44 shows the function used to calculate the Cemgil measure. In this equation, $b_a[i]$ is a list of the actual beat times and $b_p[j]$ is a list of predicted or detected beat times. The constant N_a is equal to the number of actual beats and N_p is equal to the number of predicted or detected beats. The function $W(d)$ is the weighting function and is given in Equation 45 where σ_e^2 dictates the width of the degree of match window. Cemgil et al suggest that σ_e should be equal to 0.04 sec, corresponding to the average spread of onsets from their mechanical means [14].

$$\rho(b_a, b_p) = \frac{\sum_i \max_j W(b_a[i] - b_p[j])}{(N_a + N_p)/2} \times 100\%$$

Equation 44

$$W(d) = \exp(-d^2 / 2\sigma_e^2)$$

Equation 45

While this function provides a reasonable all-in-one measure of performance, it suffers from a variety of limitations. Firstly, it does not allow for differences in metrical level between the actual and predicted beats. As stated earlier, humans frequently differ in the perception of which metrical level they feel represents the beat and so it is reasonable to believe that the proposed system is likely to differ from the expert in this regard. For example, if the actual beat is found to have frequency, or tempo, *f*, perfectly detected beats at frequency *f/2* or *2f* – a 100% valid beat prediction – can achieve a maximum Cemgil measure of only 66%. The Cemgil measure unfairly limits the acceptable metrical level of the predicted beat to precisely that dictated by the expert.

A second limitation of the Cemgil measure is its sensitivity to phase deviation. A very common beat detection error is to detect beats 180 degrees of out phase from the actual beat locations. Essentially, this means that the system has found the off-beat as opposed to the on-beat. This phase error can also be found in human beat detection (tapping, clapping or stomping) to some songs including many Country and Western and Polka songs that have a strong back-beat rhythm. In the case of an off-beat beat detection, the Cemgil measure will usually report that 0% of the detected beats are correct. While this may technically be true, it is somewhat unfair to penalize this detection to the same, possibly greater, degree as a completely random prediction of beat locations.

Since it is impossible to include auditory results for subjective evaluation in this written work, quantitative evaluations must be relied upon for an indication of the degree of success of the proposed system. If the Cemgil measure is used exclusively,

the performance of a system may appear to be significantly lower than a subjective analysis may reveal. For this reason, other options must also be considered.

6.1.2 Goto and Muraoka's Performance Measure

Goto and Muraoka approach the difficult problem of quantitative analysis and comparative beat detection system evaluation in their 1997 paper [35]. Like Cemgil et al, Goto and Muraoka compare a system's beat prediction with subjectively handpicked beat locations. However, they recognize that there are a few types of errors typically made by beat detection systems that should not necessarily be penalized. These errors include half-tempo or double-tempo errors in which the detected beat is at half or double the "correct" tempo, and π-phase errors in which the detected beats lie on the off-beats or directly between two actual beats. As stated earlier, humans commonly make both of these error types, with tempo errors being the more frequent variety. Goto and Muraoka's measure accounts for these typical errors such that systems that make these errors are not penalized.

Goto and Muraoka propose a very simple performance measure. If $b_a[n]$ are the times of actual beats, and $b_p[m]$ are the times of predicted or corrected beats, the normalized deviation error is given by Equation 46. A beat $b_p[m]$ is considered matched to an actual beat $b_a[n]$ if the predicted beat is the closest prediction to the actual beat and is within one-half of the beat period on either side of the actual beat. All actual beats without a matching predicted beat are labelled "unpaired". $P[n]$ gives the deviation error for the n^{th} correct beat [35].

$$P[n] = \begin{cases} \dfrac{\left| b_p[m] - b_a[n] \right|}{I[n]} & \text{if } b_a[n] \text{ is paired} \\ 1 & \text{if } b_a[n] \text{ is unpaired} \end{cases}$$

where,

$$I[n] = \begin{cases} \dfrac{b_a[n+1] - b_a[n]}{2} & \text{if } b_p[m] \geq b_a[n] \\ \dfrac{b_a[n] - b_a[n-1]}{2} & \text{if } b_p[m] < b_a[n] \end{cases}$$

Equation 46

The time duration τ is calculated as the longest correctly tracked period by finding the longest interval for which $P[n]$ is below some threshold and in which there are no unpaired beats. This threshold is subjectively selected to be equal to 0.35 [35]. During this period τ, deviation error mean, max, and standard deviation can be found and used as performance indicators.

This measure is easily extended to account for half-tempo, double-tempo, and π-phase errors. Since there are six possible combinations of the occurrences of these errors (tempi equal to half, unchanged or double, and phase equal to zero-phase or π-phase) the longest correctly tracked period, τ, can be calculated for each set of beat locations corresponding to one possible combination. The tempo/phase pair with the largest value of τ is selected as the best matched performance measure.

Goto and Muraoka's measure is fairly rudimentary but it does account for some of the more common beat detection system errors. By considering these errors, the measure ensures that the performance indicative values found for each set of results is more representative of the system's actual performance and not just its pure deviation from a particular set of hand-picked beat locations. However, this measure only provides information on a small stretch of well-matched beat predictions while ignoring errors occurring elsewhere in the prediction. For this reason, Goto and Muraoka's performance measure is not suitable for exclusive use in the analysis of the proposed beat prediction system.

6.1.3 Performance Measure Approach Taken

Neither the Cemgil measure nor the Goto and Muraoka measure alone is sufficient for proper analysis of a beat prediction system. That being said, the Cemgil measure can act as a limited overall system performance indicator and therefore is used to perform an initial cursory analysis in Section 6.2.1. In this section, both the proposed model and Scheirer's implementation will be analysed using the Cemgil measure. An attempt is made to minimize the effect of half and double-tempo errors by using the metrical level of the actual beat that best matches the detected beat.

The performance of the system as a whole can be estimated by finding the weighted average of the Cemgil measure across all songs where the weights are given by the number of actual beats in each song.

Equation 47 provides the formula for calculating this average where the subscript *k* denotes song number *k*. The remaining symbols in this equation are the same as those found in Equation 44 and Equation 45.

$$\tilde{\rho}\left(b_a, b_p\right) = \frac{\sum_k \sum_i \max_j W\left(b_{k,a}[i] - b_{k,p}[j]\right)}{\sum_k \left(N_{k,a} + N_{k,p}\right)/2} \times 100\%$$

Equation 47

While the Cemgil measure gives a high-level performance indication, it is instructive to analyze the specific behaviour of the proposed system, leading to a second analysis technique. There are many measures of a beat prediction system's degree of success that are important to analyze independently to best understand a system's strengths and weaknesses. These factors include the frequency of correct beat predictions, accuracy of those beat predictions, and number of spurious or incorrect beat predictions. The second phase of the analysis conducted in Section 6.2.2 examines these features in detail, beginning with the *MatchRate*.

The *MatchRate* or Match % can be defined as the proportion of actual beats that are matched by predicted beats. That is, what percentage of the actual beats is correctly

predicted by the system. Similar to Goto and Muraoka's definition of a match, a predicted beat can be said to match an actual beat if it is within some small time difference on either side of the actual beat. In this case, the maximum time difference was selected to be 80 ms; a value that corresponds roughly to the approximate mean human listener deviation time found by Scheirer [4]. Equation 48 shows the formula used to calculate this first performance measure where $b_a[n]$ is the list of actual beat times and $b_p[n]$ is the list of predicted beat times. The variables T_{max}, N_a and N_p are the maximum permitted deviation for a match, the number of actual beats, and the number of predicted beats respectively.

$$MatchRate = \frac{card\left\{n \middle| \exists m, \left|b_a[n] - b_p[m]\right| < \tau_{\max}\right\}}{N_a}$$

Equation 48

Another instructive statistic is the number of predicted beats that are spurious or are mispredicted. The *MispredictionRate* can be defined as the proportion of predicted beats that do not match an actual beat. Using the same notation as Equation 48, Equation 49 shows the formula for calculating the *MispredictionRate*.

$$MispredictionRate = \frac{card\left\{m \middle| \neg\exists n, \left|b_a[n] - b_p[m]\right| < \tau_{\max}\right\}}{N_p}$$

Equation 49

The third and final objective numerical measure used attempts to determine the accuracy of the correctly predicted beats. The mean and standard deviation of the error between correctly predicted beats and their corresponding actual beat locations are found and analysed.

The analysis in Section 6.2.2 will examine *MatchRate*, *MispredictionRate*, and prediction error mean and standard deviation. The results will be analysed independently and will be compared and contrasted with the results from Scheirer's beat detection system. Once again, half and double tempo errors are reduced by the

same method used for the Cemgil measure. Goto and Muraoka's common errors will be discussed as they relate to the examined features, specifically, how the tendency for the system to alternate between a set of valid hypotheses affects the performance.

Thirdly, a few of the test songs will be examined subjectively in greater detail in order to achieve a better understanding of the proposed system's performance as it compares to actual beat predictions and performance of Scheirer's model. Finally, a brief subjective discussion of the results will ensue, linking performance to musical genre.

6.2 Results and Analysis

The proposed system and Scheirer's model were tested using a corpus of 30 songs with a wide range of genres and tempi taken from commercial CDs. A representative segment of approximately 30 seconds was taken from each song and used for detection and analysis. The details of the songs used, including their genres, are listed in Appendix A. The tempo distribution of these songs is shown in Figure 14.

The first examination of the beat prediction results from the 30 test songs uses the Cemgil measure discussed in Section 6.1.1. Subsequently, an analysis of specific performance features will be executed followed by a detailed subjective analysis of a small selection of songs. Finally, an overall discussion of the results will examine the performance as it relates to song genre.

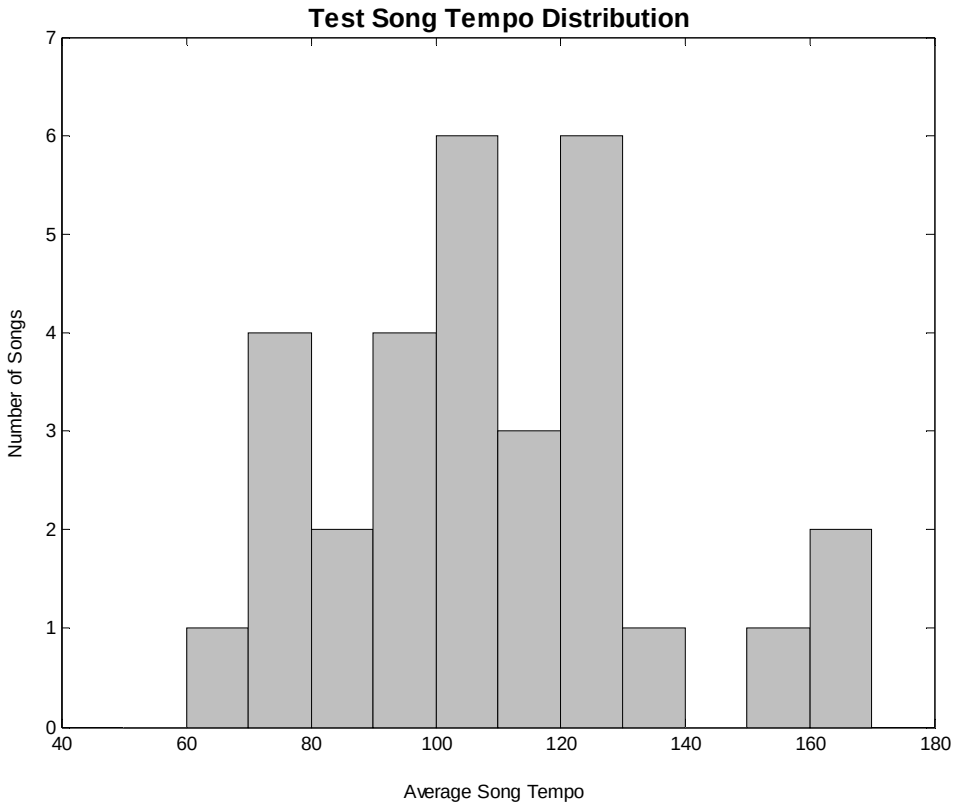

Figure 14: Song corpus tempo distribution histogram

6.2.1 Cemgil Measure Analysis

The Cemgil measure is a simple, yet effective, complete measure for analyzing the performance of beat detection systems. The calculated value is proportional to the accuracy of the predictions and the percentage of correctly predicted actual beats, and is inversely proportional to the number of spurious predicted beats. The authors also claim that it is approximately a measure of the percentage of correct beats. While this may be an ambitious claim, the measure does act as a good overall indication of detection performance.

The Cemgil measure was implemented as described in Section 6.1.1 and applied to the proposed model's predicted beat timings of each of the 30 test song segments. The rank ordered results are shown in Figure 15 where a fair distribution of results ranging from 91% to 14% can be seen. The complete numerical results can be found in Appendix B.

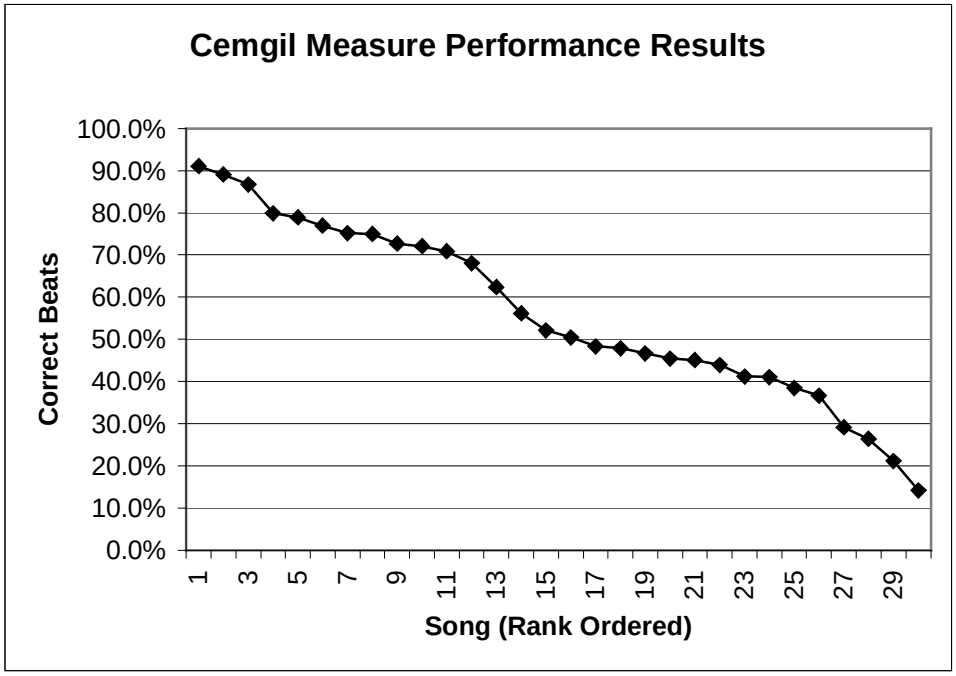

Figure 15: Cemgil measure results, rank ordered

Applying the Cemgil measure to the beat results generated by Scheirer's model using the 30-song corpus shows slightly better performance than the proposed model. Complete numerical results from Scheirer's model can be found in Appendix B. Figure 16 shows rank-ordered comparison of the results from the two models. Note that there is no direct correspondence between song rank results from each model – the results from each system are independently rank ordered. In actuality, the systems perform very differently on many music segments. In some cases, the proposed model significantly out-performs the Scheirer model, and in other cases, the reverse it true. Some of these cases will be examined in detail in a later section.

The collective performance measure for the proposed system using the specified selection of 30 songs can be found to equal 56.6% while the collective performance measure for the Scheirer system using the same corpus is calculated as 65.1%. This 8.5% performance superiority of Scheirer's model is not unexpected given the results shown in Figure 16.

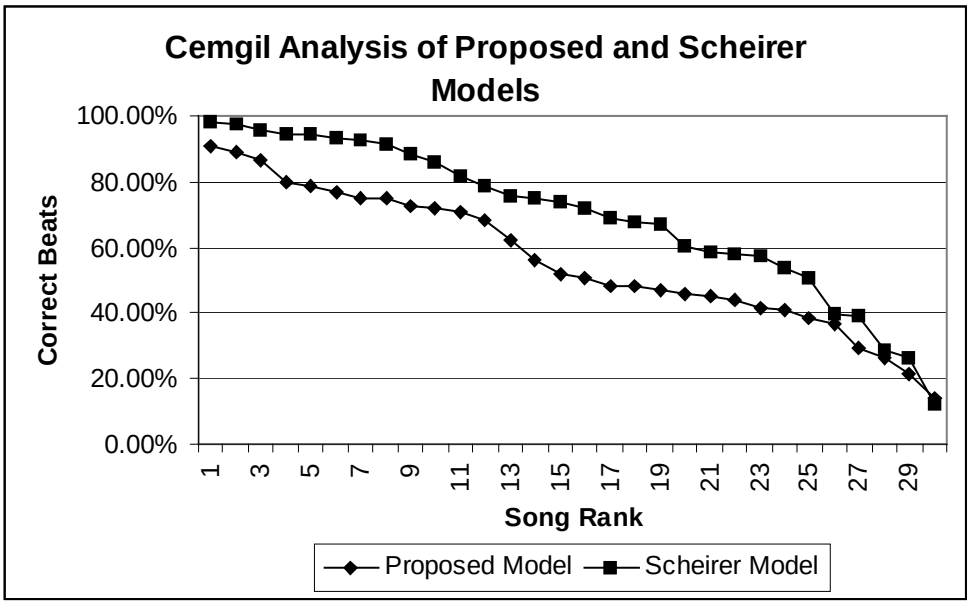

Figure 16: Cemgil measure comparative results for proposed model and Scheirer model

The Cemgil measure is an amalgam of various performance indicators and therefore it is impossible to determine the areas of strength and weakness found for each result. Furthermore, it is difficult to determine in which way the Scheirer model is out-performing the proposed model. For this reason, it is useful to consider more specific performance aspects on an independent basis.

6.2.2 Further Analysis

The primary goal of most beat prediction systems is to be able to correctly predict as many true beats as possible in the input music without incorrectly predicting additional, spurious beats. To measure a system's extent to which this goal is met, the locations of the predicted beats can be compared to the locations of the actual "expert" determined beats. One of the statistics that can be measured through this comparison is the percentage of actual beats that are correctly predicted by the system, the *MatchRate*.

Before collecting any statistic from the comparison of actual and predicted beats, the system is permitted an eight second start-up interval that allows it to learn and begin tracking the beat. This interval allows the IOI histogram to take shape and allows the recurrent timing networks to generate initial activation levels.

The rank ordered results of the *MatchRate*, the percentage of true beats that are correctly predicted by the system, are shown in Figure 17. The corresponding Cemgil measure for each song segment is also shown. It can be seen that there is a high correlation between the Cemgil measure and the *MatchRate* and that the *MatchRate* provides an approximate upper bound to the Cemgil measure. The complete numerical results for the *MatchRate* can be found in Appendix C.

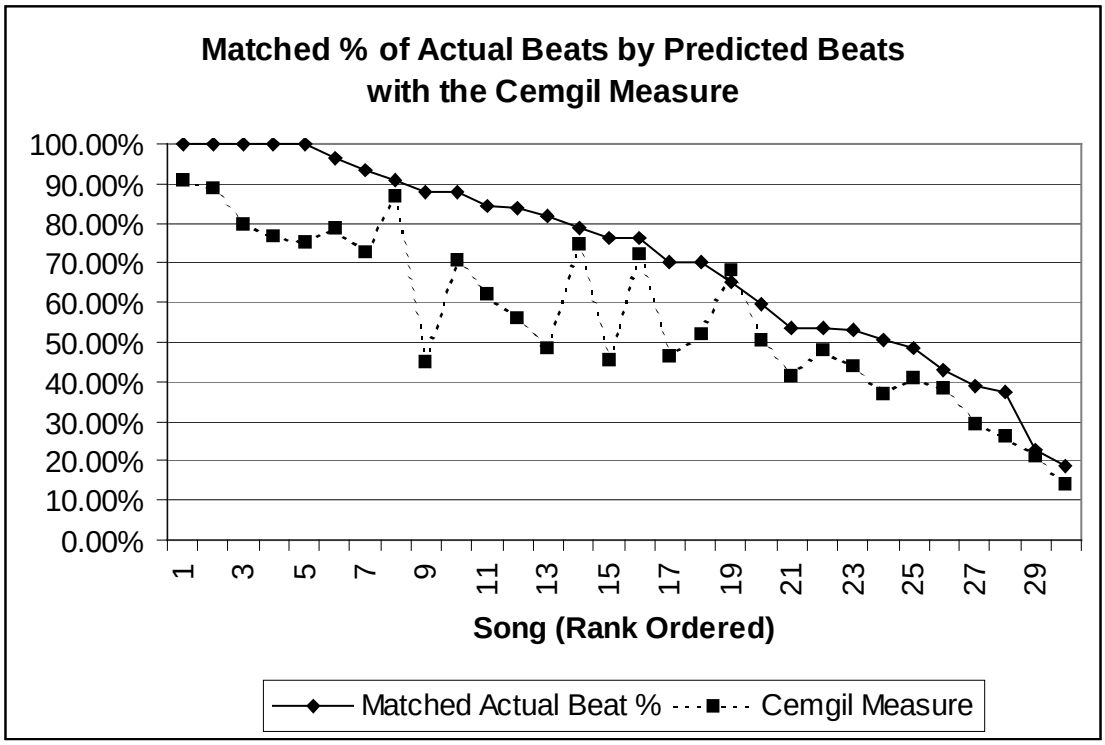

Figure 17: MatchRate with corresponding Cemgil measure

Figure 18 shows the rank ordered results of the *MatchRate* of the proposed model with the corresponding *MispredictionRate*, the percentage of incorrectly predicted or spurious beats, shown. Again, the numerical results are available in Appendix C.

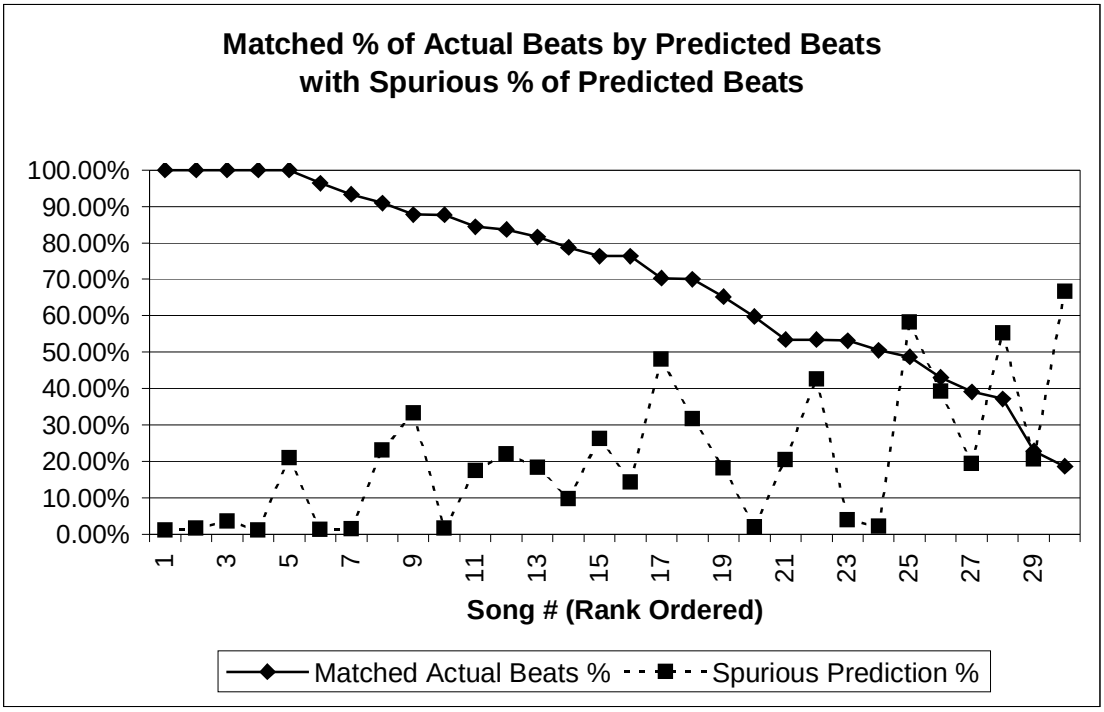

Figure 18: MatchRate with corresponding MispredictionRate

While the *MatchRate* and *MispredictionRate* shown in Figure 18 give a reasonable indication of the success of the system in predicting a song's beat, it is not always sufficient. As mentioned earlier, Goto and Muraoka identified three common errors that must be accounted for: half-tempo, double-tempo, and π-phase errors. These errors sometimes plague the entire prediction duration of a song and can be essentially cancelled out by adjusting the metrical level or phase of the "expert" determined actual beats. However, far more frequently, these errors arise for only short segments during the detection process. This is usually the result of a temporary shift to slightly stronger evidence for a new beat prediction and can result in significant reduction of most of the examined performance measures.

For example, song #29, *En Vogue's* "Hold On", is ranked #5 in Figure 18 and shows a 100% prediction *MatchRate* but suffers from a spurious misprediction percentage of 20.9%. This high spurious beat count is the result of a temporary double-tempo error for a duration of approximately 4.3 sec. These extra detected beats (detected off-beats) account for seven out of nine detected spurious beats. Therefore,

while the results report 20.9% spurious beats, only 4.6% are actually spurious. This brief tempo jump is not a desirable behaviour but the extra spurious beats are not at all random and should not be penalized as such.

The other common error type identified by Goto and Muraoka is a phase shift error and can account for some of the sub-optimal results seen in Figure 18. It is common for a beat detection node to temporarily receive stronger evidence indicating that the off-beat is stronger than the on-beat. In this case, the node may begin to generate beat predictions that are out of phase by 180 degrees. This can be seen in the case of song #3, *Fats Domino's* "Kansas City", ranked #18 in Figure 18. This song reports a 70% *MatchRate* and a 32% *MispredictionRate*. However, the last 13 predicted beats in this song were placed on the off-beat – in fact, this song has a strong swing rhythm and the predicted off-beat is not actually 180 degrees out of phase as a result, the off-beat is "swung" and lies closer to the following on-beat. This results in a large reduction in the *MatchRate* and accounts for nearly all of the mispredicted beats. If this switch did not occur, the system *may* have had a *MatchRate* of 100% and a spurious *MispredictionRate* of only 2.4%. Again, this behaviour is undesirable but should not be penalized to the degree that appears in the above figures. In fact, in accounting for these errors, the overall performance of the entire system appears to be much better than what is reported here.

The complete numerical results for the *MatchRate* and the *MispredictionRate* of the Scheirer model can be found in Appendix D. There appears to be little to no correlation between the *MatchRate* and *MispredictionRate* of the proposed model and that of the Scheirer model. However, it is interesting to note that Scheirer [4] reports of similar difficulty in system beat predictions alternating between two relevant beat hypotheses. Just as the actual performance of the proposed system appears to be much lower than its potential, Scheirer's model is likely suffering from similar effects.

Examining Figure 18 further shows that as the correct match rate decreases, the percentage of predicted beats that are spurious, on average, increases. This is expected since the system is designed to always generate a best-guess beat prediction. There is

no attempt made to cease beat prediction output when the prediction is weak. For this reason the number of predicted beats stays relatively unchanged when the system has trouble detecting a solid beat. The result is that the number of matched actual beats drops while the number of mispredictions rises.

This is not an entirely desirable behaviour and it would be useful to have an indication of how confident the system is in its prediction. The strength of a beat prediction can be inferred from the activation level of the neuron corresponding to the detected beat in the recurrent timing network of the winning node. If the average beat strength is compared with the actual beat *MatchRate*, as in Figure 19, it can be seen that large average beat strength is usually coincident with high prediction match rates. In this way, the confidence of the system in its prediction can be usually be approximated by the output beat strength.

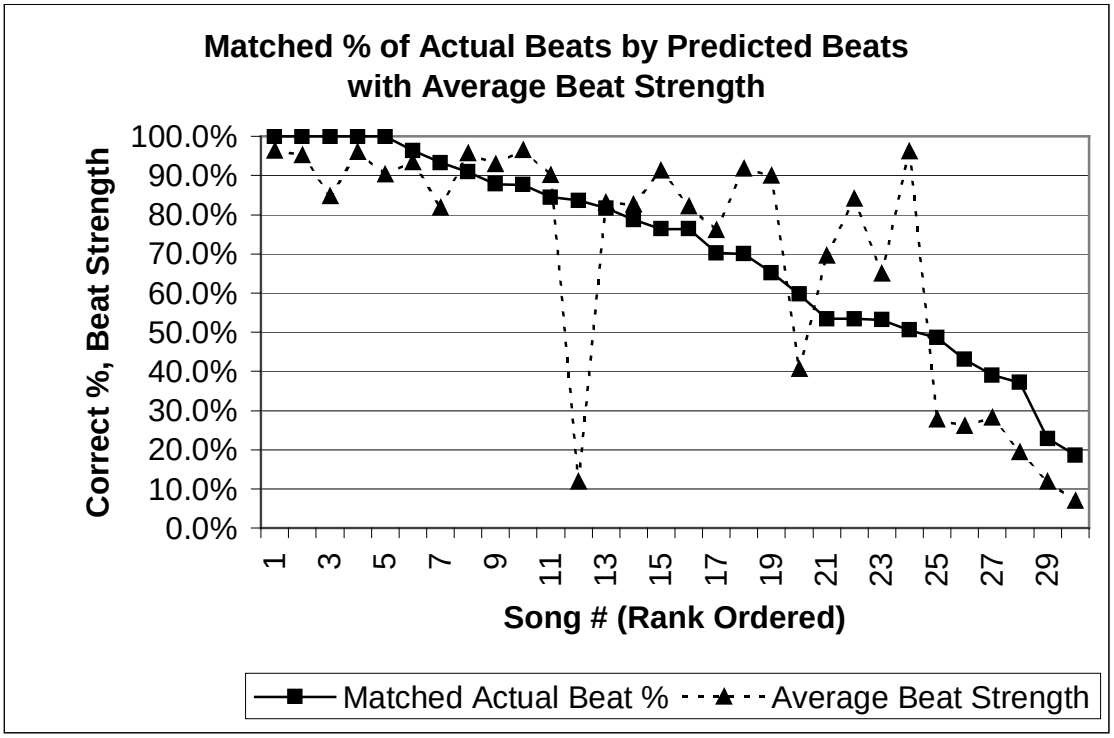

Figure 19: MatchRate with corresponding average beat strength

So far, the performance measures given in this section do not address the accuracy of the correctly predicted beats. In most applications, it is important for the predicted beats to be as close to the actual beats as possible. In the Cemgil analysis,

this accuracy measure was found in the form of a degree of match concept but was not available for independent scrutiny. In the most recent analysis, a liberal tolerance of 80 ms was used to deem a predicted beat as a match. While this was sufficient for the purpose of finding the *MatchRate*, it is instructive to know how accurate these matches actually are. To accomplish this, the error between the predicted beats and their matching actual beats is calculated.

The mean and standard deviation of the error between predicted beats and their matching actual beats was found and plotted in Figure 20. The mean is plotted with one standard deviation shown on either side. The bar graph behind the error plot shows how many beats were used for the calculation of each value and gives an indication of the validity of the error measurements. For nearly all error calculations in Figure 20, there are a sufficient number of beat deviation measurements to assume reasonably accurate calculations can be made. Appendix E contains the full numerical results.

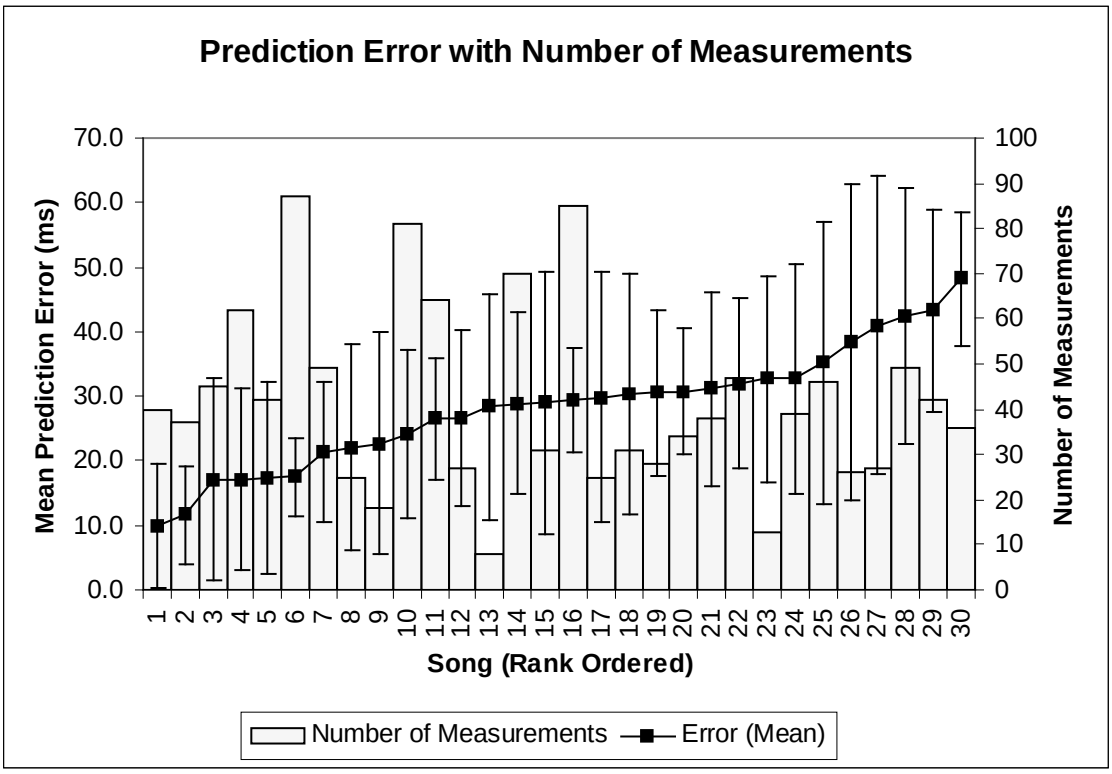

Figure 20: Prediction error mean and standard deviation with corresponding number of measurements

From Figure 20 it can be seen that the average error between predicted beats and their matching actual beats ranges from 10 to 48 ms with an overall corpus average of 28 ms. This shows a predicted beat deviation well within the acceptable levels of average human performance, however to confirm this statement, human trials would be required. The corpus average of mean deviation error in the Scheirer model predicted beats is significantly lower than the proposed model, recording 19.3 ms mean error. Since this measure is not affected by half-tempo, double-tempo, or π-phase errors, it can be interpreted to show that the Scheirer model is superior in its ability to accurately place correctly predicted beats.

Figure 21 shows the rank ordered prediction deviation error of the proposed model with corresponding *MatchRate*. It can be seen that there is no correlation between prediction error and the proportion of actual beats predicted correctly. This lack of correlation is expected. It should be noted, however, that the "expert" determined actual beats are likely to differ from the true beat of the music by some error as well. As a result, the given error results may be higher or lower depending on the accuracy of the "expert" beat locations. Once again, the fact that the actual beat location remains a perceptual concept must be not be overlooked.

At this point it is worth examining some of the results from particular song segments in more detail. This should permit a better understanding of the specific behaviours of the proposed system and its performance with respect to Scheirer's beat detection model.

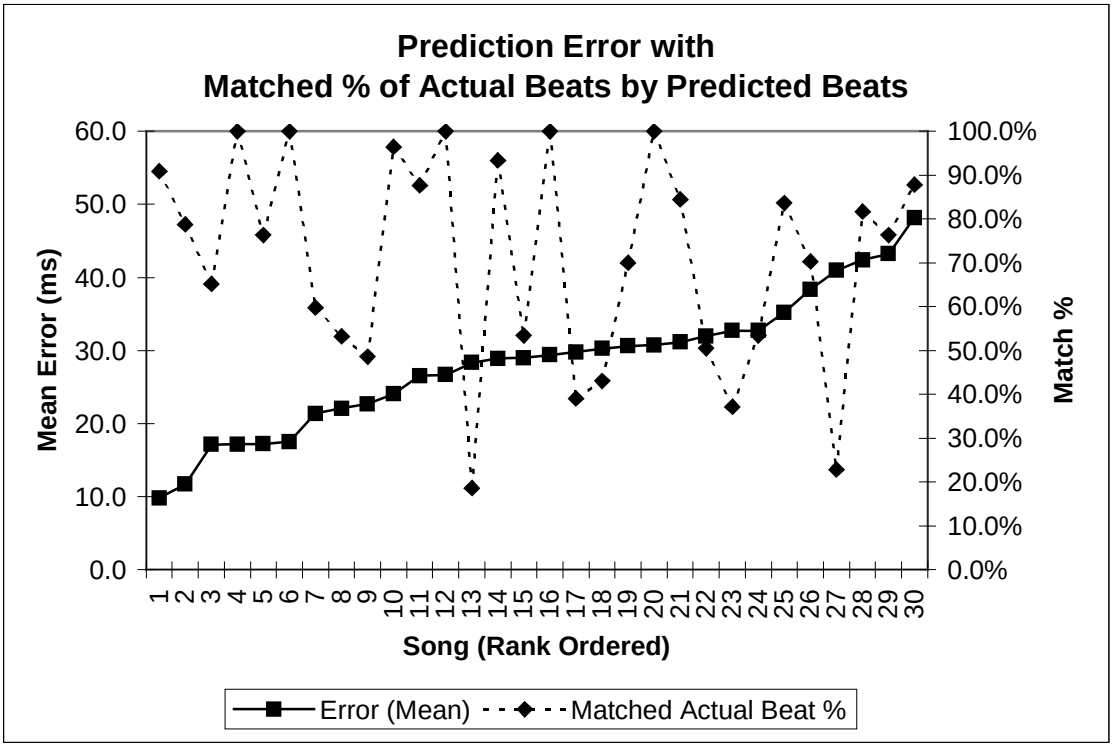

Figure 21: Prediction error with corresponding MatchRate

6.2.3 Subjective Analysis

The final analysis of the proposed system involves a brief subjective examination of the results from three different song segments. Each segment exhibits different phenomena, each discussed in previous sections, but worth examining in greater detail. The three song segments under consideration are song #15, "Take Five", song #19, "The Thrill Is Gone", and song #4, "Superstition".

6.2.3.1 Subjective Examination 1: Take Five

The focus of the first subjective examination is a segment from *The Dave Brubeck Quartet's* "Take Five". This song was selected deliberately because of its interesting *5/4* time signature. This means that each bar in this song contains five beats, contrary to the apparently stringent rules of the musical generative theory crafted by Lerdahl and Jackendoff [5]. While the generative theory specifies that the period of each metrical level must be a multiple of two or three of the level below it, "Take Five"

has no *regular* metrical level between the beat and bar levels. The bar level's period is five times the period of the beat metrical level below it.

Since the proposed beat prediction model uses minimal musical knowledge and the multiple of two or three rule was abandoned as explained in Section 5.3.2.5, this song's interesting time signature should not directly affect the model's performance. However, a consequence of this is that the model is not expected to generate an output on a reasonable metrical level – an output metrical level between the beat and bar levels is a distinct possibility. This is a potential problem in all test cases but is particularly evident here as can be seen by examining Figure 22.

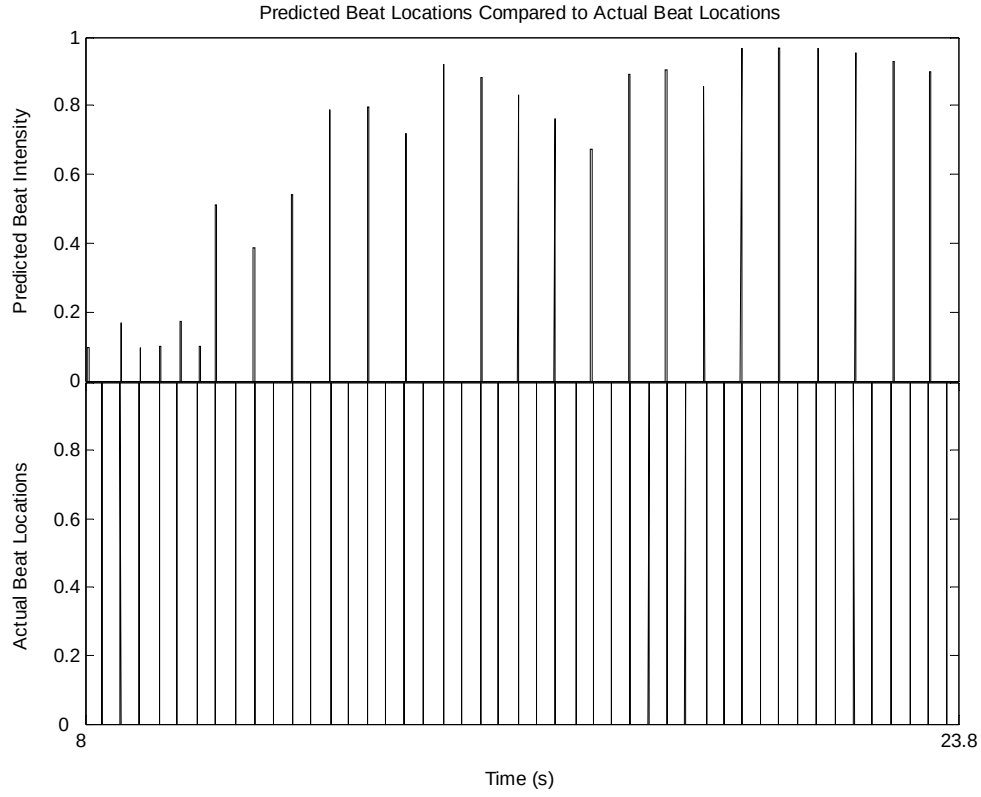

Figure 22: Predicted and actual beat locations for "Take Five"

Figure 22 shows the locations of the predicted beats above the locations of the actual beats for easy comparison. What is immediately apparent is that a beat is predicted for every other actual beat with excellent accuracy. In a song with time signature *4/4*, it is possible that this may simply represent a higher metrical level, but in

this case, this beat pattern lies between two valid levels. Beats are predicted coincident with actual beats in the sequence (where the first number represents the bar and the second number represents the beat): 1.1, 1.3, 1.5, 2.2, 2.4, 3.1, 3.3, 3.5, 4.2, … As such, the predicted metrical sequence has a period of two measures. Note, however, that all predicted beats are correct and are very close to their corresponding actual beat times, in fact, the mean error is only 22 ms.

The stream of actual beats shown in Figure 22 represents the best matching metrical level to the stream of predicted beats. Consequently, the results achieved from applying the Cemgil measure or determining the *MatchRate* report poor performance. For this segment, the *MatchRate* is only 53% – a value that belies the actual beat finding performance of the model in the case of this song segment. Of course, this stream of predicted beats is quite undesirable for use in many real-time synchronization applications since the first beat in each bar is only predicted 50% of the time! This result is an unfortunate side-effect partly from using no additional musical knowledge in the model, and partly from predicting beats in a song with an atypical time signature. Once again however, the proposed model performs much better than the quantitative results would indicate.

The Scheirer model is able to generate a better beat prediction output than the proposed model. The *MatchRate* for Scheirer's model is 72.3% – a result significantly better than the proposed model. Examining Figure 23 shows that the beat prediction made by the Scheirer model is much less stable and more erroneous than the prediction made by the proposed model. As a result, even though the Scheirer model reports a better *MatchRate* (and a much better Cemgil measure, 71.8% for the Scheirer model vs 43.9% for the proposed model), the prediction by the proposed model is arguably better. Consequently, the purely quantitative measure examined in previous sections cannot be relied upon exclusively to discern which model (proposed or Scheirer) shows truly superior performance.

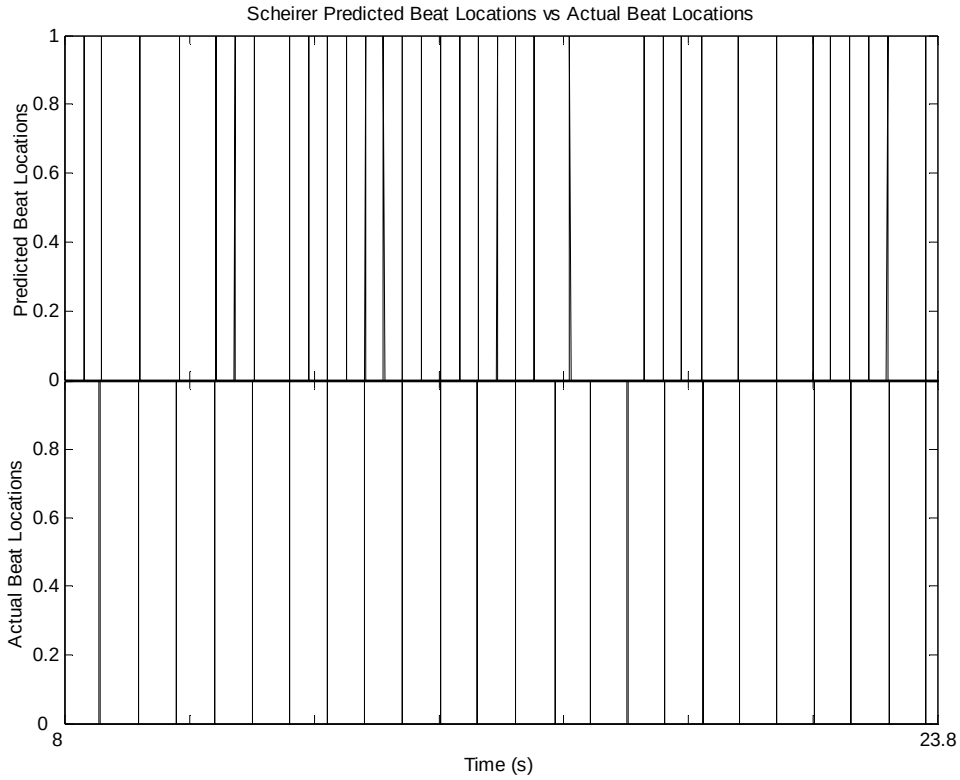

Figure 23: Scheirer model predicted and actual beat locations for "Take Five"

6.2.3.2 Subjective Examination 2: The Thrill Is Gone

This second detailed analysis assesses the performance of the model under erroneous conditions. Figure 24 shows a comparison of the predicted and actual beat locations for song #19, *B.B. King's* "The Thrill Is Gone." The highlighted portion of the figure shows a sequence of incorrect beat predictions due to an inability of the model to correctly track the beat in this section. It is apparent that the model is failing over this time period as the intensity, or confidence, of the predicted beats steadily declines. Under closer examination, it is possible to see that the predicted beats begin to drift away from the true beats starting just before the highlighted area. The system is able to recover at the 15-second mark and continue to perform well thereafter.

In order to determine the cause of this sudden beat prediction drift, it is necessary to examine the series of onsets from which this beat detection and prediction is made within the system. Figure 25 shows a comparison of the predicted beat locations, the onset locations, and the actual beat locations in this region of interest.

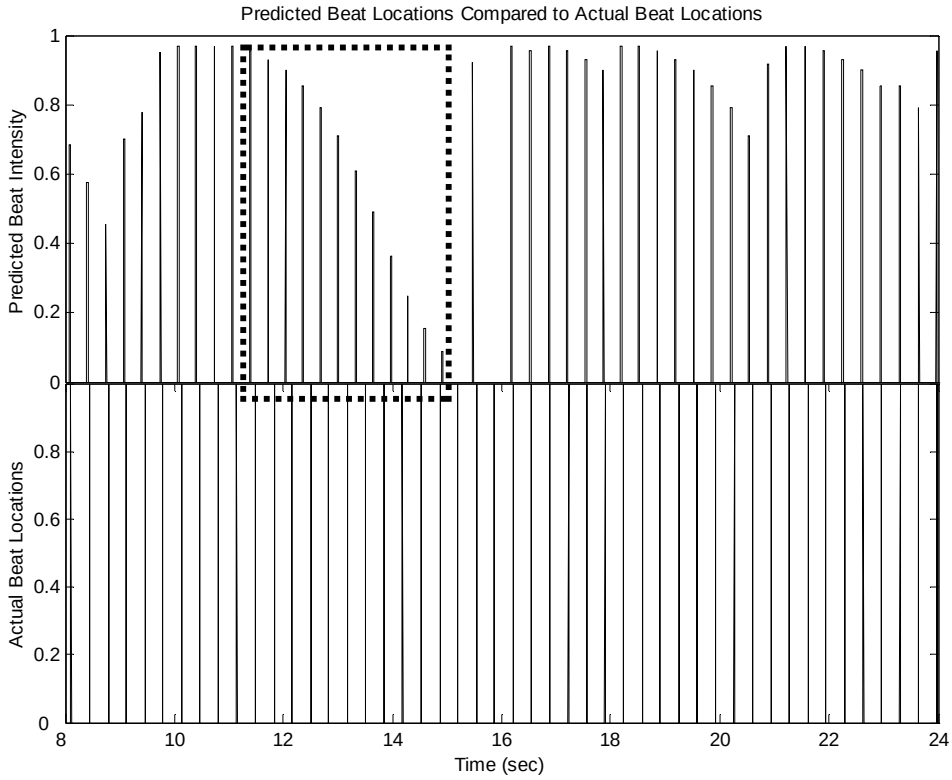

Figure 24: Predicted and actual beat locations for "The Thrill Is Gone"

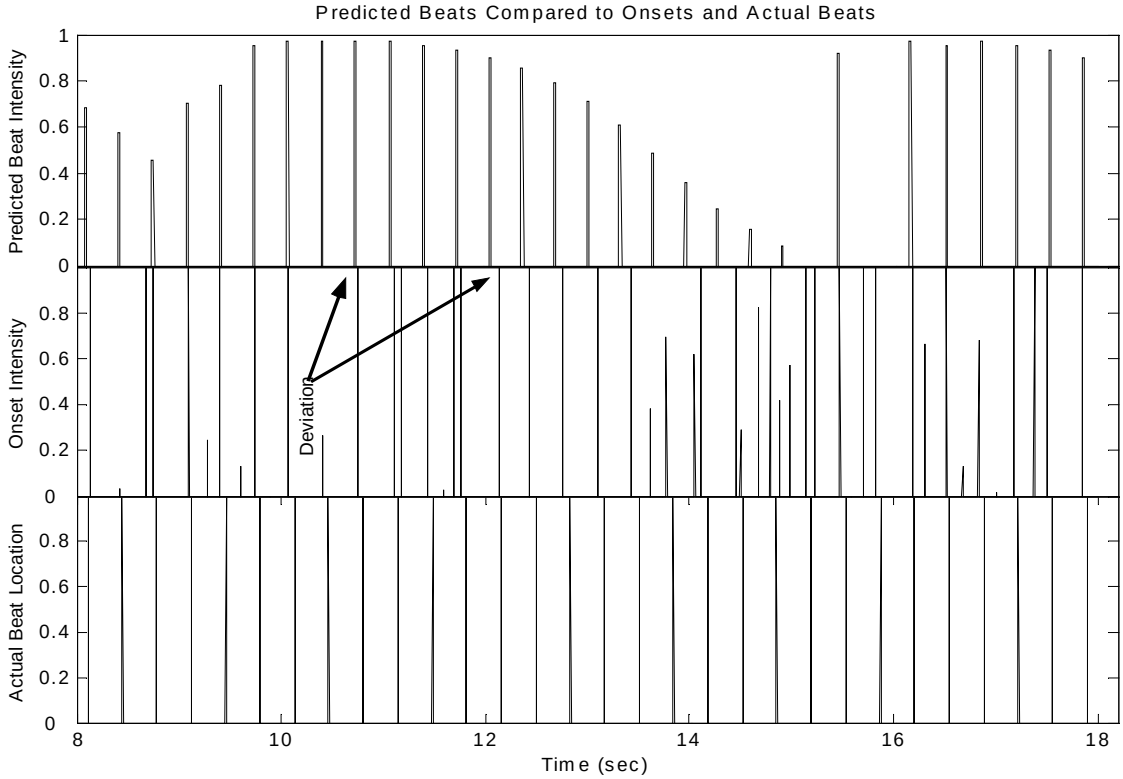

Figure 25: Predicted beats, onsets, and actual beats from "The Thrill Is Gone"

Upon examination of Figure 25, a variety of interesting information presents itself. Note the location of the left-most arrow in the figure and how the predicted beat appears to suddenly be far too early as compared to the closest onset – an onset most likely residing on the beat. The two following beats also exhibit the same trend. Because these three beats are so far from the predicted beat, they are not permitted to have much effect on the predicted beat period. This is the beginning of a serious problem.

The next beat is where the problem is exacerbated when two onsets lie closely on either side of the predicted beat. This fourth beat seems to come too late (the first of the two onsets is the closest onset) but still close enough to the predicted beat to be taken seriously. At this point, the system believes that the beat period is too long and proceeds to contract it. The result is that this particular beat detection node in the system has entered an unrecoverable error state and by the time indicated by the second arrow the deviation is far too great to be corrected by the variable rate sampler – such a large deviation causes the system to reasonably assume that the closest onsets are not actually on the beat. From this point forward, the predicted beat is incorrect and rapidly loses activation energy. Four seconds later, however, another node in the system locks on properly and continues correct beat prediction for the rest of the song.

It is unfortunate that this type of error occurs as a result of a fairly simple stream of onsets such as that shown in Figure 25. The erroneous onsets may be an artefact of the actual music or may be a result of a weakness in the onset detector. Regardless, a well-behaved system should be able to handle this type of error. The foremost problem in this test case is that a series of three onsets that indicated that the predicted beat was arriving too early was ignored and a single onset indicating the beat was too late was allowed to affect the beat period so readily. The three indicative onsets were ignored because they were too far from the predicted beat. However, logically, since there were no other onsets in the vicinity *and* all three indicated the same early beat error, they should probably have been given more weight than they were. It is difficult to say whether this change in period update logic is possible to implement without the introduction of other ill effects, but it should nonetheless be investigated.

One final point of interest with regards to this example is apparent when comparing the locations of the actual beats with the location of the closest onset. The closest onset to each actual beat is not very close at all. This is a result of either poor "expert" beat detection or poor onset detection. It is likely that both sources of error are to blame. Not surprisingly, the mean error between predicted and actual beat times for this song segment ranks among the worst in the entire corpus, at 42.4 ms. Since the system can do no better than find the beat from the information given in the onset stream, this large prediction error is, unfortunately, expected. Not surprisingly, the Scheirer model also suffers from a large mean error of 32.0 ms.

6.2.3.3 Subjective Examination 3: Superstition

The final focus for the subjective evaluation of the proposed system is song #4, *Stevie Wonder's* "Superstition". Examining the match results in Appendix C, it can be seen that even though this song has a strong, well-defined beat, the proposed system is unable to track the beat with a high degree of reliability – the *MatchRate* is only 65.2%. Figure 26 shows the locations of the predicted beats as compared with the locations of the "expert"-determined actual beats. An interesting trend is immediately apparent: the predicted beat oscillates between the correct and half-tempo. As a result, a large number of actual beats are not predicted and the *MatchRate* suffers.

This oscillatory behaviour is fairly common in the proposed model. It occurs when two beat period hypotheses are equally strong and the system is unable to decide between them. As a result, it selects one and when the other gains a small advantage, the system re-evaluates its selection. The behaviour exists in many beat detection systems. For example, Scheirer also describes the occurrence of this problem in [4], however, his model does not experience this error in this test case. Scheirer's model performs very well in this case, with a 100% *MatchRate*. There are a variety of methods for reducing the occurrence of this undesirable phenomenon that will be briefly discussed in the next chapter. However, once again, the system's beat prediction performance is arguably unjustly penalized by the selected quantitative measures.

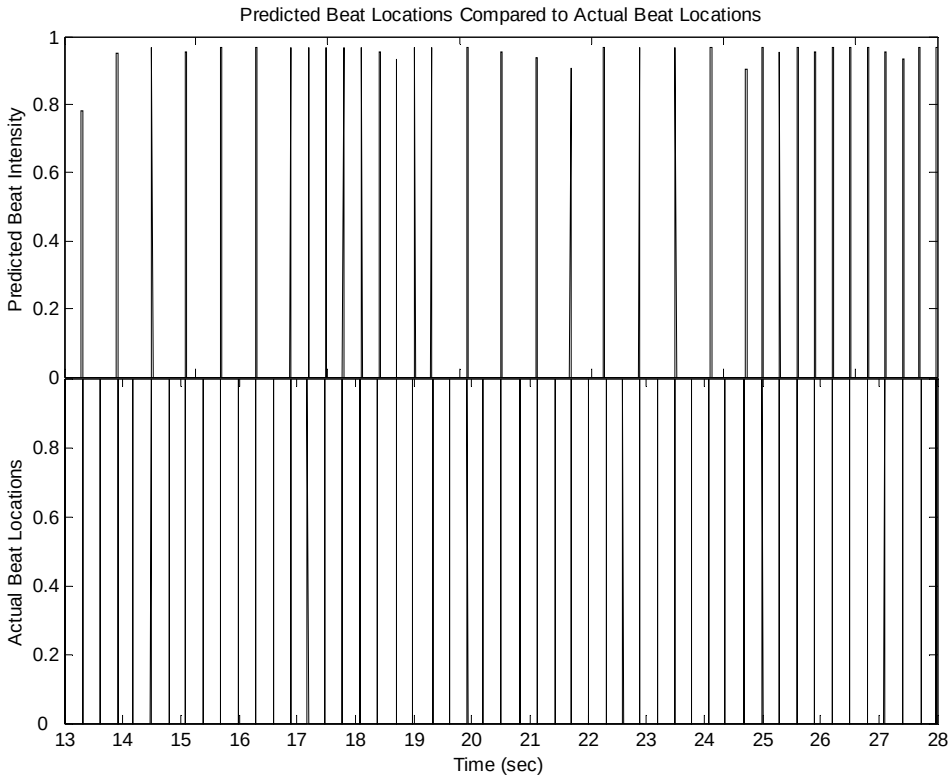

Figure 26: Predicted and actual beat locations for "Superstition"

6.2.4 Overall Analysis

Finally, before moving on, a cursory and general discussion of the results seen above is warranted. The proposed system is intended to be able to robustly predict the beat in songs with a wide variety of genres, tempi, rhythmic characteristics, and complexities. Over the 30-song corpus, a total of 1797 true beats exist, out of which, 1252 or 69.7% were correctly predicted. A significant portion of the actual beats that were missed were due to a shift to a higher metrical level in which usually only 50% of the true beats were predicted for some period of time. This result can be compared with a 72.9% correct prediction rate found using the Scheirer model. On average, 20.8% of predicted beats from the proposed model were incorrect or spurious, and again, many of these spurious beats were simply located on the off-beat. In fact, only about 12% of predicted beats were truly spurious, lying neither on the beat nor on the off-beat. In comparison, the Scheirer model generated spurious beats on average of 17.8% of the

time. Based on these statistics, the proposed model can be considered reasonably successful and comparable to the Scheirer model.

The performance of the system was not consistent across all genres. Some genres, such as rock music, contained songs in which the beat was easier to find, whereas other genres contained songs with significantly more rhythmic complexity. Table 3 summarizes the performance of the system on a per genre basis using the *MatchRate* to divide results into good (70-100%), moderate (40-69%), and poor (0-39%) categories.

Table 3: Summary of performance results by genre

Genres	Songs	Successful (70-100%)	Moderate (40-69.9%)	Poor (0-39.9%)
• Alternative • Classic Rock • Hard Rock	8	6	2	0
• Classic R&B • Modern R&B • Hip Hop • Blues	6	4	1	1
• Country	2	2	0	0
• Dance • Electronica • Industrial	6	3	3	0
• Easy Listening • Instrumental • Vocal	4	3	0	1
• Jazz	2	0	1	1
• Classical	2	0	1	1
Totals	**30**	**18**	**8**	**4**

Not surprisingly, the jazz and classical genres suffer from the worst performance, most likely due to the complexity of the music, the rhythm, and subtly of the beat. This being said, "Take Five" actually performs quite well as seen in Section 6.2.3.1 but is unfairly penalized due to the nature of the evaluation, resulting in a

"Moderate" ranking. Also not surprising is the poor performance in the vocal genre due to the very subtle presence of rhythm normally found in this type of music.

One surprising result is the poor match rates found in dance and electronica genres, usually known for their strong rhythmic content. A number of the songs in these genres, namely the songs by *Daft Punk* and *The Crystal Method* achieved very high match rates (over 90%), whereas the pulsating rhythm of the song by *Aqua* performs only moderately with a 50% match rate. Under closer examination, the *Aqua* song is found to suffer from a π-phase error for nearly the entire duration of the song. Had this error not occurred, the match rate would likely have been near 100% as expected for a song in this genre. Not surprisingly, the Scheirer model is able to perfectly detect the beat in *Aqua's* song in the corpus, avoiding the proposed model's π-phase error.

In summary, the performance of the proposed model appears to be quite good across most genres. Eliminating oscillation between different metrical levels of the beat and reducing π-phase errors would greatly improve the success rate of many of the song segments under consideration.

Chapter 7: Conclusions and Recommendations

7.1 Recommendations

Two major difficulties in the proposed system exist and have been reiterated a number of times in this document. These problems are the tendency for beat output metrical level oscillation and the frequent existence of π-phase errors. Upon the correction of these two undesirable features, the performance of the proposed model and its viability for continued research and application would be vastly improved.

Reducing the tendency for the system to oscillate between two metrical level hypotheses is a rather straightforward task. This may be able to be solved with no more work than adding logic specifying that the system should pick a hypothesis and "stick to it". As long as the beat intensity of a prediction stays high, the prediction output is most likely still valid and does not need to be reconsidered. Only switching to a new hypothesis if the current hypothesis is outscored significantly and fails to remain a valid prediction should help alleviate this prolific problem.

Eliminating many of the π-phase errors may also have, in concept, a simple solution. In the current model, one of Lerdahl and Jackendoff's more important rules regarding metrical levels is being ignored. While the integer period ratios of different metrical levels in the current system are used to aid in node score calculation, the rule that the beats of a higher metrical level should always coincide with beats from a lower metrical level is overlooked. When selecting the activation energy spike within a recurrent timing network to represent the beat, a particular node could examine all nodes that represent higher metrical levels. Decisions as to the location of the beat should be made such that it is consistent across all metrical levels. This would result in beat hypotheses that are supported, not only in period, but also in phase by many of the beat hypothesis nodes in the system. This increased cooperation could greatly benefit the performance of the entire system.

Taking the above approach one step further, the period adjustment behaviour of the system could be improved and its susceptibility to error could be reduced. Instead of simply using other metrical level representations from other nodes in the system to bolster a node's score, these other nodes could aid in period adjustment. Since metrical levels should always have coincident beats and integer ratio periods, period adjustment should ensure that modifications made are appropriate across all metrical levels and these integer multiple periods are maintained. In this way, period adjustment using inter-node cooperation could greatly enhance the system's robustness and performance.

7.2 Conclusions

The purpose of the research presented here was to propose a new approach for the prediction of the beat in an acoustic musical signal. An updated version of the basic recurrent timing network was used to form the basis of the periodic element detection crucial to any beat prediction system. Independent, self-adjusting agents, each containing such a network, were employed to find possible beats and compete for activation as the node representing the best beat hypothesis. This complex, interacting system was designed to robustly predict the beat using new techniques intended to expand and improve the field of real-time automatic beat prediction.

The model was tested using a collection of song segments from a wide variety of musical genres. The overall performance of the model using this test set was moderate to good with true-beat prediction rates averaging near 70%, with only 12% of predictions being incorrect on average. The overall measure proposed by Cemgil et al showed a total correct beat percentage of near 57%, a value well in line with other acoustic beat prediction systems.

Overall, a large proportion of songs from a wide range of genres showed favourable results. The proposed system is quite evidently not confined to predicting beats only in music with a pulsing bass drum or prominent drum tracks as performances in genres such as instrumental and easy listening music show. While the model's success rate in jazz and classical music is somewhat poor, it is not unreasonable, as

these genres frequently have beats that are more subtle and rhythmic structures that are more complex than most other types of music.

While in general, the system performed well, it suffered from a variety of drawbacks. First and foremost, the model had a tendency to oscillate between two different metrical levels of the same beat hypothesis throughout the course of a song segment. This phenomenon was examined in detail and found not only to adversely affect the performance measures, but also to limit the efficacy of the output for use without post-prediction manual adjustment.

Another problem that plagued the system was a propensity for detecting the off-beat instead of the desirable on-beat. Again, this resulted in decreased performance both in theoretical measure and in real-world applicability. Both the oscillatory tendency and permission of π-phase errors should be corrected to bring this model closer to real world application viability.

The ultimate goal of this beat prediction research was to create a different approach to beat prediction that would provide a promising direction for future research in this area. While the proposed beat prediction system suffers from a variety of undesirable behaviours, its overall performance is nonetheless promising and warrants further research effort.

Appendix A: Corpus Information

#	Genre	Artist	Title	Album
01	Alternative	Billy Talent	Try Honesty	Billy Talent
02	Alternative	Cake	Short Skirt, Long Jacket	Comfort Eagle
03	Classic Rock	Fats Domino	Kansas City	Gold Collection
04	Classic R&B	Stevie Wonder	Superstition	Talking Book
05	Classic Rock	The Beatles	Lucy in the Sky with Diamonds	Sgt. Pepper's Lonely Hearts Club Band
06	Classical	Dvorák	Slavonic Dance no. 1	The Best of Dvorák
07	Classical	J.S. Bach	Brandenburg Concerto No. 2 in F Major – 3rd Movement	The Best of the Toronto Symphony Orchestra
08	Country	Tim McGraw	Where the Green Grass Grows	Everywhere
09	Dance	Aqua	Barbie Girl	Aquarium
10	Easy Listening	Phil Collins	Can't Stop Loving You	Can't Stop Loving You
11	Electronica	Daft Punk	Around the World	Homework
12	Hard Rock	AC/DC	You Shook Me All Night Long	Back in Black
13	Hip Hop	Nelly	Hot in Herre	Nellyville
14	Industrial	Kraftwerk	The Model	The Man-Machine
15	Jazz	The Dave Brubeck Quartet	Take Five	Time Out
16	Jazz	The Oscar Peterson Trio	Night Train	Night Train
17	Modern R&B	Beyonce Knowles feat. Sean Paul	Baby Boy	Dangerously In Love
18	Alternative	Sublime	Santeria	Sublime
19	Blues	BB King	The Thrill is Gone	Blues Masters 7 – Blues Revival
20	Classic R&B	The Temptations	My Girl	Ultimate Collection
21	Classic Rock	The Beach Boys	Surfin U.S.A.	Surfin' Sarafi
22	Country	Alabama	I'm in a Hurry	American Pride
23	Dance	Justin Timberlake	Rock Your Body	Justified
24	Dance	Moby	In this World	18
25	Easy Listening	Dave Matthews Band	Cry Freedom	Crash
26	Electronica	The Crystal Method	Busy Child	Vegas
27	Hard Rock	Rammstein	Du Hast	Sehnsucht
28	Instrumental	David Tolly	Endless Love	Greatest Love Themes vol. 4
29	Modern R&B	En Vogue	Hold On	Born to Sing
30	Vocal	Sarah Vaughn	Misty	The #1 Jazz Album

Appendix B: Results – Cemgil Measure

Song #	Proposed Model Cemgil Measure	Rank	Scheirer Model Cemgil Measure	Rank
1	45.4%	20	68.8%	17
2	78.9%	5	81.8%	11
3	52.1%	15	12.0%	30
4	68.1%	12	67.2%	19
5	41.1%	24	67.8%	18
6	21.2%	29	50.3%	25
7	50.4%	16	57.6%	23
8	72.1%	10	94.7%	4
9	36.7%	26	94.5%	5
10	70.8%	11	73.9%	15
11	86.7%	3	98.1%	1
12	62.4%	13	60.4%	20
13	72.7%	9	93.6%	6
14	91.1%	1	85.9%	10
15	43.9%	22	71.8%	16
16	26.4%	28	39.1%	27
17	29.1%	27	78.8%	12
18	89.1%	2	91.5%	8
19	48.3%	17	58.3%	21
20	46.7%	19	57.9%	22
21	41.2%	23	75.7%	13
22	74.9%	8	97.6%	2
23	47.9%	18	75.2%	14
24	38.5%	25	96.0%	3
25	79.9%	4	88.4%	9
26	76.9%	6	39.8%	26
27	45.1%	21	53.9%	24
28	56.1%	14	92.6%	7
29	75.2%	7	28.7%	28
30	14.2%	30	26.2%	29

Appendix C: Results – Actual Beat Match Statistics (Proposed Model)

	Actual Beats		Predicted Beats	
Song #	**Match %**	**Miss %**	**Correct %**	**Mispredict %**
1	76.4%	23.6%	73.7%	26.3%
2	96.4%	3.6%	98.8%	1.2%
3	70.0%	30.0%	68.3%	31.7%
4	65.2%	34.8%	81.8%	18.2%
5	48.6%	51.4%	41.9%	58.1%
6	22.9%	77.1%	79.4%	20.6%
7	59.8%	40.2%	98.0%	2.0%
8	76.4%	23.6%	85.7%	14.3%
9	50.5%	49.5%	97.9%	2.1%
10	87.7%	12.3%	98.5%	1.5%
11	90.9%	9.1%	76.9%	23.1%
12	84.4%	15.6%	82.6%	17.4%
13	93.3%	6.7%	98.6%	1.4%
14	100.0%	0.0%	98.9%	1.1%
15	53.2%	46.8%	96.2%	3.8%
16	37.1%	62.9%	44.8%	55.2%
17	39.1%	60.9%	80.6%	19.4%
18	100.0%	0.0%	98.4%	1.6%
19	81.7%	18.3%	81.7%	18.3%
20	70.3%	29.7%	52.0%	48.0%
21	53.4%	46.6%	79.5%	20.5%
22	78.7%	21.3%	90.2%	9.8%
23	53.4%	46.6%	57.4%	42.6%
24	43.1%	56.9%	60.8%	39.2%
25	100.0%	0.0%	96.4%	3.6%
26	100.0%	0.0%	98.8%	1.2%
27	87.8%	12.2%	66.7%	33.3%
28	83.6%	16.4%	78.0%	22.0%
29	100.0%	0.0%	79.1%	20.9%
30	18.6%	81.4%	33.3%	66.7%

Appendix D: Results – Actual Beat Match Statistics (Scheirer Model)

	Actual Beats		Predicted Beats	
Song #	**Match %**	**Miss %**	**Correct %**	**Mispredict %**
1	81.5%	18.5%	71.0%	29.0%
2	90.5%	9.5%	90.5%	9.5%
3	10.0%	90.0%	9.8%	90.2%
4	100.0%	0.0%	94.4%	5.6%
5	86.5%	13.5%	84.2%	15.8%
6	61.0%	39.0%	67.9%	32.1%
7	78.0%	22.0%	76.2%	23.8%
8	100.0%	0.0%	81.8%	18.2%
9	100.0%	0.0%	100.0%	0.0%
10	83.3%	16.7%	69.8%	30.2%
11	100.0%	0.0%	100.0%	0.0%
12	71.1%	28.9%	78.0%	22.0%
13	100.0%	0.0%	100.0%	0.0%
14	88.4%	11.6%	100.0%	0.0%
15	72.3%	27.7%	100.0%	0.0%
16	45.7%	54.3%	42.1%	57.9%
17	81.3%	18.8%	78.8%	21.2%
18	72.6%	27.4%	97.8%	2.2%
19	80.0%	20.0%	77.4%	22.6%
20	70.3%	29.7%	68.4%	31.6%
21	93.1%	6.9%	96.4%	3.6%
22	100.0%	0.0%	97.9%	2.1%
23	100.0%	0.0%	94.7%	5.3%
24	100.0%	0.0%	100.0%	0.0%
25	100.0%	0.0%	73.0%	27.0%
26	40.0%	60.0%	94.4%	5.6%
27	39.8%	60.2%	100.0%	0.0%
28	83.6%	16.4%	100.0%	0.0%
29	32.4%	67.6%	100.0%	0.0%
30	41.9%	58.1%	36.0%	64.0%

Appendix E: Results – Beat Prediction Accuracy (Proposed Model)

Song #	Error Mean	Error Standard Deviation	Number of Measurements	Rank
1	43.2	15.7	42	29
2	24.0	13.0	81	10
3	30.6	12.8	28	19
4	17.1	15.7	45	3
5	22.7	17.2	18	9
6	40.9	23.1	27	27
7	21.4	10.9	49	7
8	17.2	14.8	42	5
9	32.0	13.1	47	22
10	26.5	9.5	64	11
11	9.8	9.6	40	1
12	31.1	15.0	38	21
13	28.9	14.0	70	14
14	17.5	5.9	87	6
15	22.1	15.9	25	8
16	32.7	16.0	13	23
17	29.8	19.3	25	17
18	17.2	13.9	62	4
19	42.4	19.9	49	28
20	38.4	24.4	26	26
21	29.0	20.3	31	15
22	11.7	7.6	37	2
23	32.7	17.9	39	24
24	30.2	18.6	31	18
25	26.7	13.7	27	12
26	29.4	8.1	85	16
27	48.2	10.5	36	30
28	35.2	21.8	46	25
29	30.7	9.7	34	20
30	28.3	17.5	8	13

Appendix F: Results – Beat Prediction Accuracy (Scheirer Model)

Song #	Error Mean	Error Standard Deviation	Number of Measurements	Rank
1	23.1	9.5	22	22
2	18.9	2.9	38	16
3	23.2	5.0	4	23
4	34.6	19.9	34	29
5	25.1	19.7	32	24
6	21.0	24.2	36	18
7	28.6	19.1	32	26
8	12.6	6.9	27	8
9	13.7	5.2	46	9
10	16.3	14.3	30	14
11	8.2	3.6	44	3
12	21.7	14.9	32	20
13	15.2	4.2	37	11
14	10.3	2.8	38	5
15	21.2	16.0	34	19
16	18.6	18.7	16	15
17	10.1	4.5	26	4
18	16.0	9.9	45	12
19	32.0	19.6	24	28
20	25.4	21.2	26	25
21	22.4	22.7	54	21
22	7.5	6.5	47	2
23	31.0	6.6	36	27
24	12.3	2.5	36	6
25	16.1	16.7	27	13
26	4.2	3.7	34	1
27	12.5	6.5	33	7
28	14.4	5.8	46	10
29	20.5	3.3	22	17
30	43.6	18.7	18	30

References

[1] M. R. Jones and E. W. Large, "The Dynamics of Attending: How People Track Time-Varying Events," *Psychological Review*, vol. 106, pp. 119-159, 1999.

[2] E. W. Large, "On Synchronizing Movements to Music," *Human Movement Science*, vol. 19, pp. 527-566, 2000.

[3] E. W. Large and C. Palmer, "Perceiving Temporal Regularity in Music," *Cognitive Science*, vol. 26, pp. 1-37, 2002.

[4] E. D. Scheirer, "Tempo and Beat Analysis of Acoustic Musical Signals," *Journal of the Acoustic Society of America*, vol. 103, pp. 588-601, 1998.

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

[6] D. Moelants, "Preferred Tempo Reconsidered," presented at 7th International Conference on Music Perception and Cognition, Sydney, Austrailia, 2002.

[7] P. Fraisse, "Rhythm and Tempo," in *The Psychology of Music*, D. Deutsch, Ed. New York: Academic Press, 1982, pp. 149-180.

[8] P. Fraisse, "Time and Rhythm Perception," in *Handbook of Perception*, vol. 8, E. C. Carterette and M. P. Friedman, Eds. New York: Academic Press, 1978, pp. 203-254.

[9] J. Seppänen, "Computational Models of Musical Meter Recognition," in *Department of Information Technology*. Tampere, Finland: Tampere University of Technology, 2001, pp. 72.

[10] J. Laroche, "Estimating Tempo, Swing and Beat Locations in Audio Recordings," presented at International Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), Mohonk, NY, USA, 2001.

[11] J. Foote and S. Uchihashi, "The Beat Spectrum: a New Approach To Rhythm Analysis," presented at IEEE International Conference on Multimedia & Expo, Tokyo, Japan, 2001.

[12] D. Eck, "A Network of Relaxation Oscillators that Finds Downbeats in Rhythms," *Lecture Notes in Computer Science*, vol. 2130, pp. 1239-1247, 2001.

[13] S. Dixon, "Automatic extraction of tempo and beat from expressive performances," *Journal of New Music Research*, vol. 30, 2001.

[14] A. T. Cemgil, B. Kappen, P. Desain, and H. Honing, "On Tempo Tracking: Tempogram Tepresentation and Kalman Filtering," *Journal of New Music Research*, 2001.

[15] P. Cariani, "Temporal Codes, Timing Nets, and Music Perception," *Journal of New Music Research*, vol. 30, pp. 1-52, 2001.

[16] M. Alghoniemy and A. Tewfik, "Rhythm and Periodicity Detection in Polyphonic Music," presented at IEEE Third Workshop on Multimedia Signal Processing, Denmark, 1999.

[17] M. Goto and Y. Muraoka, "Music understanding at the beat level: Real-time beat tracking for audio signals," in *Computational Auditory Scene Analysis*, D. F. Rosenthal and H. G. Okuno, Eds. Mahwah, NJ, USA: Lawrence Erlbaum Associates, 1998, pp. 157–176.

[18] M. Goto and Y. Muraoka, "A Beat Tracking System for Acoustic Signals of Music," in *ACM Multimedia*, 1994, pp. 365-372.

[19] L. M. Smith, "Listening to Musical Rhythms with Progressive Wavelets," presented at IEEE TENCON - Digital Signal Processing Applications, 1996.

[20] D. Rosenthal, "Machine Rhythm: Computer Emulation of Human Rhythm Perception." Cambridge, MA: Massachusetts Institute of Technology, 1992.

[21] D. Povel and P. Essens, "Perception of temporal patterns," *Music Perception*, vol. 2, pp. 411–440, 1985.

[22] P. E. Allen and R. B. Dannenberg, "Tracking Musical Beats in Real Time," presented at International Computer Music Conference, Glasgow, Scotland, 1990.

[23] R. Parncutt, "A Perceptual Model of Pulse Salience and Metrical Accent in Musical Rhythms," *Music Perception*, vol. 11, pp. 409-464, 1994.

[24] P. Desain and H. Honing, "The Quantization of Musical Time: A Connectionist Approach," *Computer Music Journal*, vol. 13, pp. 56-66, 1989.

[25] P. Cariani, "Timing Nets for Rhythm Perception (Working Paper)," presented at The Symposium on Music on Timing Nets, Ghent, Belgium, 1999.

[26] K. Gurney, *An Introduction to Neural Networks*. London, England: University College of London Press Ltd, 1997.

[27] A. Klapuri, "Sound Onset Detection by Applying Psychoacoustic Knowledge," presented at IEEE International Conference on Acoustics, Speech, and Signal Processing, 1999.

[28] C. Duxbury, M. Sandler, and M. E. Davies, "A Hybrid Approach to Musical Note Onset Detection," presented at The 5th International Conference on Digital Audio Effects, Hamburg, Germany, 2002.

[29] N. P. M. Todd and G. J. Brown, "The Visualization of Rhythm, Time and Metre," *Artificial Intelligence Review*, vol. 10, pp. 253-273, 1996.

[30] J. Seppänen, "Tatum Grid Analysis of Musical Signals," presented at IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, New Paltz, NY, USA, 2001.

[31] G. C. Goodwin, S. F. Graebe, and M. E. Salgado, *Control System Design*: Prentice Hall, 2001.

[32] C. L. Krumhansl, "Rhythm and Pitch in Music Cognition," *Psychological Bulletin*, vol. 126, pp. 159-179, 2000.

[33] "ACID Pro 4.0 (Demo Version)." Madison, WI: Sonic Foundry Inc. (www.sonicfoundry.com/Acid), 2003.

[34] E. D. Scheirer, "Implementation of Tempo and Beat Analysis of Acoustic Musical Systems," http://web.media.mit.edu/~eds/beat/tapping.tar.gz ed, 1998.

[35] M. Goto and Y. Muraoka, "Issues in Evaluating Beat Tracking Systems," presented at IJCAI-97 Workshop on Issues in AI and Music, 1997.