An Automatic Blues Band

Andrew BROTHWELL
Dept. of Computer Science
University of Bath
BATH BA2 5QR,
United Kingdom

John FITCH
Codemist Ltd
Horsecombe Vale
BATH BA2 7AY,
United Kingdon
jpff@codemist.co.uk

Abstract

The development of a computer musical accompanist is presented, that attempts to model some of the characteristics of human musicianship, particularly concentrating on the human sense of rhythm, and the playing 'by ear' skill set. It specifically focuses on blues accompaniment, and works with no previous knowledge of the particular song that is about to be played. The performance of the lead musician is constantly monitored and reacted to throughout the song.

The system is written in Csound and runs in real time.

Keywords

Blues, Accompanist, Csound

1 Introduction

An issue that faces many musicians is that practising and playing their instruments on their own can become dull and unfulfilling. This is particularly true with musicians playing 'lead' instruments in styles that would usually be played as part of a full band. These can often sound empty when played alone without backing chords or accompaniment.

Currently musicians have two options if they want to play their instrument along with accompaniment. First they can join a group of musicians, that is human musical accompanists who know how to listen to the lead musician, and play in the key and the tempo that the lead musician chooses. In most cases this is the preferred choice, and will most likely be so for many years to come. However there is a second option, artificial accompaniment, which is the topic of this paper.

Artificial musical accompaniment has been around for many years in various forms, although not necessarily computer based or using any artificial intelligence. Although the versatility and intelligence of an artificial accompaniment will not match its human counterpart, there are many practical advantages; it will be available at any time, will not cost anything significant to run, will be infinitely patient, and will never tire.

At the most basic level, artificial accompaniments are compact discs containing compilations of pre-recorded music, with certain musical parts not included in the mix. This allows players of that instrument to fill in the missing part. This is arguably not accompaniment at all as it is the musician who must listen and follow the backing band. Despite this limitation this type of product is very popular with musicians, as they give at least some impression of playing with others.

In this paper we address the question of whether it is possible to develop a computer system that can accompany a lead player without a score, working solely on audio input and without specific knowledge of the song about to be played.

The desire for an artificial accompanist is not new. For example Dannenberg[1] demonstrated a system where there was a known score. Of direct input to this project was the system of Bryson[2] which predicted the correct chord to accompany a single note melody line, using a subsumption strategy. At the time it was not possible for such a system to work in real time, but it demonstrated that the prediction mechanism can be applied.

Our aim is a system that 'listens' by way of a simple microphone, and plays in time and in tune with the lead. The software base is Csound[3].

2 Requirements for Accompanist

The system's overall aim is to emulate a small twelve-bar blues accompaniment section. We can therefore extract much of what we want it to achieve by looking at what would be expected from these human blues accompanists if they were following a lead musician.

The first consideration is which instruments

we want to have in the accompanying "band". A basic blues band will typically consist of a Lead Instrument (Acoustic/Electric Guitar, Piano, Mouth Organ etc), a Rhythm Guitar, Bass Guitar and Drum Kit. The musician using the system will play the lead instrument and the computer will be responsible for playing the other instruments.

Certain elements of twelve-bar blues are relevant to all the musicians in the band. The song is divided into parts, bars, and beats. There are four beats to each bar and twelve bars to each part. Each of the twelve bars will be played with a backing chord relative to the key of the song, in the order of: I, I, I, I, IV, IV, I, I, V, IV, I, I, where we use Roman numerals to indicate the position of the chord played. For all performers we need them to listen for lead musician's count-in, and then play the four beats in each bar and the twelve bars in each part following this. While doing this they need to constantly listen to the lead musician and adjust tempo and pitch to match if necessary. At the end of each twelve bar part they need to decide if the song is going to end, and if it is then play some final fill and stop. In addition there are skills that will be relevant only to a musician playing a particular instrument.

The Drummer has to come in exactly on the first beat of the bar following the count-in, and in tempo. On the first and third beats of each bar he plays the kick drum[1]. On the second and fourth beat of each bar he plays the snare drum, and on each half beat, the hi-hat symbol.

The Bassist and Rhythm Guitarist both need to begin a blues bass line or strum exactly on the first beat of the bar following the count-in. This will be played in the pentatonic (blues) scale, and will start on the root note of the scale that the lead musician is playing in. From then they need to follow the twelve bar blues chord changes and end on a final root note.

3 The Challenges and Solutions

3.1 Pitch Detection

It is important that the synthetic players play in tune with the soloist, whatever pitch he or she uses. There are a number of possible pitch detection algorithms, in either the time or frequency domain. After some experimentation we concluded that the average magnitude difference function (AMDF)[4] was sufficiently robust.[2]

3.2 Tempo and Beat Tracking

The musical accompanist project will rely heavily on being able to detect tempo. In terms of a human, the first indication of an incompetent accompanist is that he cannot stay in time. Accurate tempo detection will give the whole system a far more realistic and professional sound. The problem of tempo detection is twofold. We have to establish the speed that the musician is playing at in beats per minute. On top of this, however, we must calculate the actual locations in time of these beats. In other words, to be able to predict when future beats will occur, we not only need to know the gaps between beats, but also the time at which the beats occurred.

The problem of onset detection is also well researched. Many onset detection algorithms rely on the bass drum for the beat, but in our case the input will have no percussion and this affects what methods will work. Csound does not have a pre-made onset detection method so there is need for some developments.

We found that a very simple onset detector, working by extracting the tempo from the rate of notes played, was sufficient forour needs. An onset is recorded, as the start of a note, at the time the root mean square of the input audio reaches a level above a certain threshold. As the input to the system is likely to be more melodic than percussive, this method depends on the instrument having peaks in amplitude when a note is first played. If we pick the right threshold level for onset detection this issue should not be critical. It does however need adjustment of the threshold.

Once notes can successfully be recognised,

[1] In blues drumming it is often popular to play it twice in quick succession on the third beat

[2] Since the completion of the basic system a frequency domain pitch tracker, **ptrack**, has become available in Csound and we intend to test that scheme as it seems to have some advantages such as the absence of parameters.

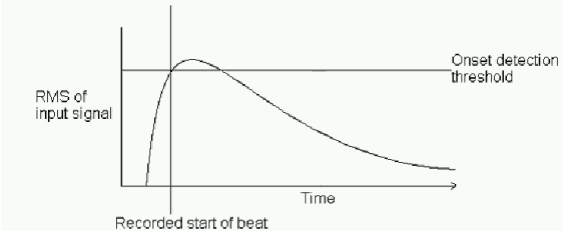

Figure 1: Onset Detection by RMS

calculating the beat length, and therefore the speed of the song, is trivial. In terms of performance this method requires little calculation, and as we will consistently be listening for the beats throughout the song this is an essential attribute.

One problem of using this method during the count-in is that the RMS of the input signal will almost certainly remain above the threshold level for a length of time while the note rings. This is not another note being played, it is simply the remainder of the note that has already been registered, and so we need to ensure that during this time multiple beats are not falsely recognised. This problem was solved by ignoring the input signal for a set time period after each of the count-in beats was recorded. This period could be calculated reasonably, as we knew the fastest reasonable tempo that a musician would play, and hence the shortest possible time in between beats.

3.3 Starting

As with a human band of accompanists, we would like the system to be triggered into starting by the musician playing a four beat count-in. This count-in is often given in the real world verbally by counting aloud to the rest of the band. It is also common for the lead musician to count-in by playing four short notes on his instrument, and in this system we will assume that this will be the case.

By using the onset detection method to register the beats in this count-in, we can calculate the beat length of the song that is about to start, and come in both at the right time and at the right speed. There is a need to compensate for latency in the system. After the last beat in the count-in, the average length of a beat is calculated. This provided information about the tempo, and latency in the system had no profound effect on this. It was in the determining of the beat positions that latency came into play — it is one thing to play at the right speed, but to play in time requires beats to be synchronised exactly. A latency constant was used to allow for this delay, and simply subtracted from the calculated beat position. The value of this latency constant was calculated mainly through trial and improvement; it was fairly easy to judge when the accompanist started too early or too late. It is worth noting here that computer systems with different hardware, operating systems, and compiler versions

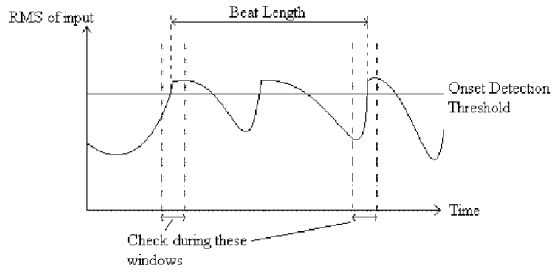

Figure 2: First Beat Correction Algorithm

will almost certainly require this constant to be recalculated, and it was set by another of the system calibration variables at the top of the program code.

3.4 Staying in time

It would also be preferable to continue tracking the speed that the musician is playing at throughout the song. The system should then be able to continually calculate the tempo, and react accordingly to it. This will allow the system both to correct any error in its initial tempo belief, and to stay in time with the musician should he speed up or slow down.

This aspect of the program was the most problematic. Initially we experimented with an algorithm that expected an on-beat note within a certain window and checked for it, as described in figure 2 and as pseudo-code as

```
If (Time = WindowStart AND
    InputRMS > Threshold) //already ringing
 GotFirst = False
 STOP CHECKING TILL NEXT WINDOW
ElseIf (InputRMS > Threshold AND
        WindowStart < Time < WindowEnd)
 If(GotFirst)
  SecondBeat = Time
  BelievedBeatLength = SecondBeat - FirstBeat
  FirstBeat = SecondBeat
  STOP CHECKING TILL NEXT WINDOW
 Else
  FirstBeat = Time
  GotFirst = True
  STOP CHECKING TILL NEXT WINDOW
 EndIf
ElseIf (Time = WindowEnd)
 GotFirst = False
EndIf
```

This algorithm, while not affected by latency can suffer from persisting in a belief that the timing is correct while it slowly drifts away. As a result an alternative method was developed, which needs fine tuning, but is more responsive to adjustments.

The idea is look at the predicted time and make micro adjustments if the onset is not when expected.

```
If (Time = WindowStart AND
    InputRMS > Threshold) // ringing
 STOP CHECKING TILL NEXT WINDOW
ElseIf (WindowStart < Time < WindowEnd
        AND InputRMS > Threshold)
 RealBeatTime = Time
 TempoError = RealBeatTime -
              BelievedBeatTime
 STOP CHECKING TILL NEXT WINDOW
EndIf
```

Making use of the corrected timing is not obvious. Direct use gives an unnatural jumping in the backing music which breaks the flow and feel of the song. After analysing some human accompanists it was realised that as a lead musician speeds up, the accompanying musicians will not instantaneously react and adjust exactly to the new tempo, but rather will do so gradually. The initial attempt to model this effect was to use a constant fraction to multiply by the tempo error by when adding it on (a value of around 0.3 worked best).

This meant that the accompanist system approached the new tempo gradually over a few beats rather than all in one go. This sounded much more realistic with the song continuing to flow far more smoothly. There still seemed to be an absence of a real human sense of rhythm though, and after further analysis of human accompaniment the difference was found.

Human accompanists do not directly alter the tempo they are playing at, but rather, in a manner of speaking, adjust their acceleration. That is to say that they alter their rate of tempo change. This means that they will change tempo faster if the tempo has already been changing for the previous few bars, and they will continue to change until they approach the lead musician's new tempo where they will gradually reduce the rate of tempo change until

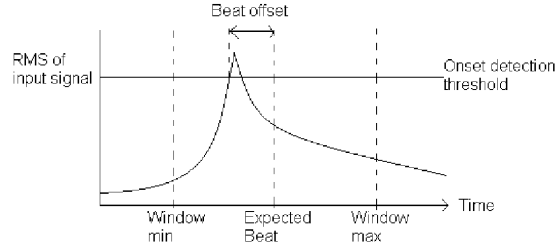

Figure 3: Second Beat Correction Algorithm

they again match him exactly. By doing this, the song never loses its rhythm and feel, rapid tempo change can occur, but it must be built up to. There have been many analyses of performance, for example the work of Sundberg([5] and earlier papers) or Clarke[6], which give scientific weight to our simple observations.

What we need is to add the tempo error not directly to the believed beat length, but rather to a separate variable storing overall tempo change. Then this is added to the believed beat length every beat. In this way the tempo change variable in effect represented the rate of tempo change, or acceleration, and just like with a human musician it was this that was adjusted rather than the tempo directly. Another observation about human accompaniment was that tempo changes for the most recent few beats have far more influence than tempo changes from several beats beforehand. With this method so far, tempo changes from the entire song had equal bearing, which was obviously not right. What we needed was to multiply the tempo change variable by a constant fraction each beat – in effect a simple filter. After some experimentation it was found that halving the variable each beat produced great results, and the accompanist now changed speed smoothly, and sounded far more realistic and with a human sense of rhythm.

3.5 Staying in Tune

In development it was found that the system would sometimes play very slightly out of tune after the count-in. Although it was only by a very small amount, the system was very slightly sharp when an acoustic guitar was used as an input.

The problem was not in the pitch detection but a physical one with the way the guitar was played. When counting in using the root note, short notes were being played, and the strings were being plucked quite hard. When playing melodies during the song, however, plucking was gentler to suit a more flowing style.

A string that is plucked hard will have a higher pitch at first due to the extra tension caused by the string deflecting further. This is noticeable for instance in the "twang" made by a plucked elastic band. It is not usually noticeable in a guitar as it is so short lived. This system however was tuning all the other instruments to the pitch of that initial hit of the string. What was needed was to take the

pitch detected during the count-in, but to refine this belief whenever the musician played another root note during the song. The issue here was to identify when the musician was playing a root note.

The closest note that a musician is likely to play to the root note will be one semitone away either higher or lower. Even this is fairly unlikely as the notes a semitone away are not part of the blues scale, but to be safe we say that if we detected a pitch that was less than half a semitone (3%) away from the believed root note, then it is likely that the musician is playing a root note. In this case we will update the believed root note to be the average of the current belief and the note just played. This is the method implemented to fine-tune the pitch during the song. In this way the system's belief was updated every time that the lead musician played a root note. Any very slight errors throughout the song were gradually reduced.

3.6 Stopping together

Knowing when an improvised song is going to finish is amongst the most abstract skills that human accompanists will have. The only initial knowledge that they have is that the song is very likely to finish at the end of a twelve-bar part. It turns out to be very hard to define the "feeling" that the song is about to finish, although this is definitely something that is present even in people who are not musicians. It can sometimes be a change in the way the notes flow, sometimes a change in tempo, sometimes a change in volume, or a combination of any of these. This is certainly a part of music that is difficult to analyse scientifically.

During a practise with one of the authors' band, an attempt was made to ascertain the processes that musicians go through when listening for an ending. It was clear that hearing was not the only sense used by the band. Often the body language of the musicians, and eye contact made were equally important.

This was verified when the band tried to play with their eyes closed; they lacked their usual tightness and had moments of uncertaint, especially the transitions between loud and soft parts which were a challenge. They did end together however showing that listening is the main clue[3]

Once we have successfully recognised an appropriate time to end, we must also have an appropriate way to end. Simply stopping sounded very weak, as though somebody had just turned a record off, and was very unnatural.

Live blues endings can be very complex and drawn out, and these longer endings are often entirely improvised. One can sometimes feel as if musicians use the ending as an escape from the constraints of the rest of the song and an opportunity to show off their most impressive off the cuff playing. Improvisation like this is beyond this system. It is in fact perfectly acceptable, however, for an accompanying band to play a far simpler ending. A single drum hit with a sustained root note from the guitar and bass invariably sounds good, and so this method was chosen to be implemented.

[3]Some of the greatest blues musicians of all time were completely blind, so this is obviously the case!

4 Conclusions

The system has been tested by a few blues musicians, and the overall reaction has been favourable. The instruments were justly criticised but the pitch and tempo reactiveness was praised. More details can be found in [7]

The project met its original aim: the system itself is fun to use, and it is used by us when practising and experimenting.

The Csound source is available in the technical report[7] and on the Dream web site[8].

References

[1] Roger Dannerberg. An on-line algorithm for real-time accompaniment. In *Proceedings of the International Computer Music Conference*, pages 193–198, Paris, 1984.

[2] Joanna Bryson. The subsumption strategy development of a music modelling system. Master's thesis, Department of Artificial Intelligence, University of Edinburgh, 1992.

[3] Richard Boulanger, editor. *The Csound Book: Tutorials in Software Synthesis and Sound Design*. MIT Press, 2000.

[4] G. S. Ying, L. H. Jamieson, and C. D. Michell. A probabilistic approach to AMDF pitch detection. In *Proc. ICSLP '96*, volume 2, pages 1201–1204, Philadelphia, PA, 1996.

[5] A. Friberg, R. Bresin, and J. Sundberg. Overview of the KTH rule system for musical performance. *Advances in Cognitive Psychology, Special Issue on Music Performance*, 2(2-3):145–161, 2006.

[6] Eric Clarke. Rhythm and timing in music. In Diana Deutsch, editor, *The Psychology of Music*, pages 473–500. Academic Press, San Diego, revised edition, 1999.

[7] Andrew Brothwell. Developing an artificially intelligent real time blues accompanist. Technical Report CSBU-2007-01, Department of Computer Science, University of Bath, March 2007. available from `http://www.cs.bath.ac.uk/department/technical-series-database/technical-report-dissertation.html`.

[8] Andrew Brothwell. Acompanst. `http://dream.cs.bath.ac.uk/software/acompanist.csd`, 2005.