

# RHYTHMIC ANALYSIS FOR REAL-TIME AUDIO EFFECTS

Adam M. Stark, Matthew E.P. Davies and Mark D. Plumbley

Queen Mary, University of London  
Centre for Digital Music  
adam.stark@elec.qmul.ac.uk

## ABSTRACT

We outline a set of audio effects that use rhythmic analysis, in particular the extraction of beat and tempo information, to automatically synchronise temporal parameters to the input signal. We demonstrate that this analysis, known as beat-tracking, can be used to create adaptive parameters that adjust themselves according to changes in the properties of the input signal. We present common audio effects such as delay, tremolo and auto-wah augmented in this fashion and discuss their real-time implementation as Audio Unit plug-ins and objects for Max/MSP.

## 1. INTRODUCTION

The use of Audio Effects is becoming increasingly common in live performance and studio recordings. Nevertheless, some types of effect can be quite challenging to achieve successfully. For example, effects which rely on the tempo or rhythm information, such as a one-beat delay (echo), can be quite difficult for a musician to apply, particularly when playing live.

Using a commercial software sequencer, such as Logic Studio (Apple, Inc.), the musician would synchronise to a click track to ensure their playing remained synchronised with their pre-set “one beat” delay. However, the musician cannot deviate from the set tempo and this can lead to a performance seeming mechanical or inexpressive. Alternatively, the musician could use a delay foot pedal, such as the Echo Park (Line 6, Inc.), tapping the current tempo to set the delay. However, if the tempo changes during the performance, the musician must concentrate on regularly updating the tempo value set in the pedal.

Our approach is to use the music signal itself to apply the effect, through the use of beat-tracking. Prior work on such signal-dependent effects includes Compressors and Noise Gates that monitor the input level and process the signal if it passes a certain threshold [7, pp100-104]. Features such as pitch have also been used to control the parameters of audio effects [6]. Here we use beat-tracking analysis of the signal, which has also been successfully used in live synchronisation with a drummer [3].

## 2. AUDIO EFFECTS

Let us briefly review some common digital audio effects that we will modify with our beat tracking system.

The *delay* effect [7, p63], similar to an echo, causes a signal to be repeated after a certain amount of time and mixed with the original signal. It is implemented by:

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

where  $x[n]$  and  $y[n]$  are the input and output signals,  $\alpha$  is the gain level of the delayed signal and  $Q[n]$  is the delay in audio samples set by the user.

The *tremolo* effect [7, p77] is the amplitude modulation of a signal by a low frequency oscillator (LFO) and is implemented by:

$$y[n] = x[n] \cdot m[n] \quad (2)$$

where  $x[n]$  and  $y[n]$  are the input and output signals and  $m[n]$  is the LFO waveform operating at a user-defined frequency (Hz).

The *auto-wah* effect [7, pp55-56] is the modulation of the center frequency,  $f_c$ , of a band pass filter by a LFO and is implemented by:

$$f_c = f_b + (m[n] \cdot f_r) \quad (3)$$

where  $f_b$  is a base frequency for the bandpass filter in Hz,  $f_r$  is the sweep range in Hz and  $m[n]$  is the LFO waveform operating at a user defined frequency (Hz).

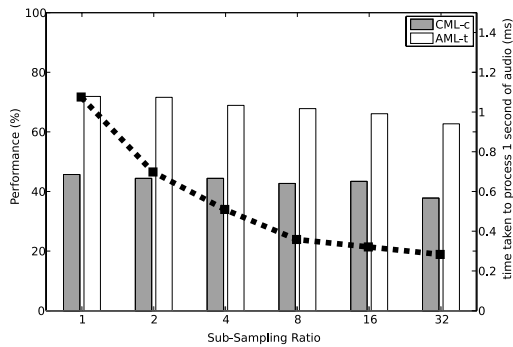
The *flanger* effect [7, pp69-71] is created by processing a signal using a variable delay line modulated by a LFO and mixing the result with the input signal:

$$y[n] = x[n] + \alpha \cdot x[n - D[n]] \quad (4)$$

where  $x[n]$  and  $y[n]$  are the input and output signals,  $\alpha$  is the amount of the delayed signal that is mixed with the original and  $D[n]$  is the variable delay length calculated by:

$$D[n] = m[n] \cdot T_{max} \quad (5)$$

where  $m[n]$  is the LFO operating at a user-defined frequency (Hz) and  $T_{max}$  is the maximum delay time.



**Figure 1.** The bar graph shows the performance of the beat tracker using different levels of sub-sampling. The dotted line shows the amount of time needed (in ms) to process one second of audio for the same levels of sub-sampling. There is a large decrease in computation time without significant loss in performance as the level of sub-sampling increases.

### 3. BEAT TRACKING

#### 3.1. Beat Tracking System

The beat-tracker employed in this paper is an implementation of the system presented in [1], although it should be noted that the effects described in this paper are designed to work with any beat-tracker. A brief description of this system follows.

The input signal, sampled at 44.1kHz, is split into 512 sample audio frames and transformed using an FFT. The spectral difference between consecutive frames is then observed to create a detection function (DF) output. After 128 DF samples have been observed ( $\sim 1.5$  seconds) they are placed into a 512 DF sample buffer ( $\sim 6$  seconds) with the previous 384 DF samples ( $\sim 4.5$  seconds). The auto-correlation function is then calculated on this detection function before a comb-filter is used to extract the time between beats, or beat period. Finally, the most recent samples of the detection function relating to a single beat period are analysed to extract the beat alignment. Using this information, beats are predicted into the future for the next  $\sim 1.5$  seconds after which the analysis process is repeated for the duration of the input.

#### 3.2. Baseline Evaluation

The beat tracker was tested on the Hainsworth [2] library of 222 audio files lasting collectively for over 3 hours. The beat-tracker was tested in order to provide some objective measure of the performance of the beat and tempo-synchronous audio effects. Two measures, used previously to evaluate other beat-tracking systems [1, 2], were employed to assess the performance of the beat tracker. The first, Correct Metrical Level with continuity required (CML-c), calculates the longest correctly tracked continuous section of audio as a percentage of the whole file. The second, Allowed Metrical Levels with continuity not required (AML-t), calculates the number of correctly tracked

beats as a percentage of the total and allows beat tracking at double or half the correct tempo and on the off-beat. For the CML-c the beat-tracker performed at 45.7% while for the AML-t it performed at 71.9%. These results are comparable to other state of the art causal beat trackers. For more information on the metrics employed here and further results on this database please refer to [1].

#### 3.3. Speeding Up Analysis

While the initial implementation of the beat tracker worked comfortably in real-time (averaging 1ms to process 1 second of 44.1kHz audio), it was found that considerable improvements in speed could be achieved by downsampling the audio before performing the FFT. This may be desirable if the system was implemented using hardware with less computational power. The system was tested with the audio sub-sampled by 2, 4, 8, 16 and 32. Figure 1 shows the results of this testing. As can be seen, the performance of the beat-tracker is relatively unaffected by sub-sampling while the computation time is significantly decreased. This implies that the most salient events related to the beat occur at lower frequencies. As a result of this we suggest that sub-sampling audio by a rate of approximately 8 is desirable to achieve better computational efficiency while maintaining performance. The reduction in processing time as the audio is downsampled shows that the majority of the computational load is in the calculation of the onset detection function and not the beat tracking element.

### 4. SYNCHRONOUS AUDIO EFFECTS

We define here the term *tempo-synchronous* to refer to the synchronisation of parameters of an audio effect to the tempo of the input signal. This is done using the beat period information from the beat tracker. Beat alignment information is not required for tempo-synchronous effects. We define *beat-synchronous* effects to be audio effects that have parameters synchronised to the input signal using both beat period and beat alignment information.

#### 4.1. BeatDelay: A Tempo-Synchronous Delay

We create a tempo-synchronous delay effect by replacing the length of the delay in samples,  $Q[n]$ , in equation (1) by a value related to the beat period,  $\tau[n]$ , by a number of beats,  $\lambda$ :

$$Q[n] = (\lambda \cdot \tau[n]) \quad (6)$$

This tempo-synchronous delay allows easy creation of delay effects that are related to the tempo of the input signal and adaptive over time.

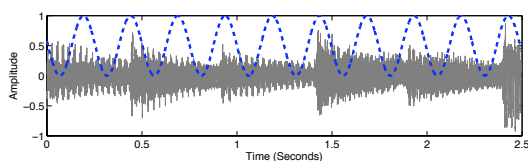
#### 4.2. Beat Synchronous Low Frequency Oscillator

All the beat-synchronous audio effects described here are based upon a beat-synchronous low frequency oscillator

(LFO). By beat-synchronous, we mean that the cycle length of the LFO is related to the beat period by some integer value and that the LFO is ‘in phase’ with the beats. This is achieved by starting the LFO at a beat-location and adjusting the length of LFO cycles so that they are related to the tempo of the input signal by a number of *cycles per beat* (see Figure 2). Two different forms of the beat-synchronous LFO were implemented, one for situations where the LFO operates at one or more cycles per beat (*type-I*) and one for situations where LFO cycles are more than one beat in length (*type-II*). Type-I LFOs were implemented by setting the rate of the oscillator at each beat to cause it to reach a peak at the next beat after performing  $\Omega$  cycles per beat. Type-II LFOs were implemented by adjusting the rate of the oscillator at each beat, mid cycle, to deal with tempo changes and thus to cause it to end exactly  $\psi$  beats after it started, where  $\psi = 1/\Omega$  and  $\Omega < 1$  (e.g 0.5, 0.33, 0.25, etc).

#### 4.2.1. LFO Phase Mismatches

The causal and predictive nature of the beat-tracker allowed the real-time implementation of the audio effects described in this paper. However, it also led to difficulties in the implementation of effects based upon the beat-synchronous LFO. As the beats are predicted at the end of each  $\sim 1.5$  second frame, there were problems at the last beat of each frame as at this point, it was impossible to know the location of the first beat of the next frame and therefore to correctly set the rate of the LFO for the next beat. The solution to this problem was to use corrective algorithms between the first and second beats of the next frame to correct for any errors in the phase of the LFO. Type-I LFOs may end slightly before or after the first beat and so at the first LFO peak after the first beat a corrective rate was set to cause it to end exactly at the second beat. For Type-II LFOs problems only occurred if the cycle was due to finish on the first beat of a frame. If the LFO finished early and therefore had begun another cycle before the first beat, the LFO rate was set so that it reached the correct point at the second beat. If the LFO had not finished by the first beat then it was allowed to both complete its cycle and reach the correct phase between beats 1 and 2. This technique is discussed in more detail in [4].



**Figure 2.** A Beat-Synchronous LFO operating at 2 cycles per beat. Beats occur at approximately 0.45, 0.9, 1.35 and 1.8 seconds.



**Figure 3.** BeatDelay - A tempo-synchronous delay effect.

### 4.3. Beat-Synchronous Effects: BeatTrem, BeatWah and BeatFlanger

To implement the beat-synchronous Tremolo, Auto-Wah and Flanger, we simply set  $m[n]$  in equations (2), (3) and (5) respectively to the LFO from the beat tracker:

$$m[n] = LFO_{\Omega}[n] \quad (7)$$

where  $\Omega$  is the number of cycles per beat. The parameter  $\Omega$  is set by the musician from an interface control panel, as part of an Audio Unit or in Max/MSP, but could also be set from a hardware control.

Because of the requirements of the particular effects, the LFO for the Tremolo was restricted to one or more cycles per beat, the Auto-Wah to one cycle per beat and the flanger to a rate of one or more beats per cycle,  $\psi$ , i.e.  $\Omega = 1/\psi$ .

### 4.4. Real-Time Implementation

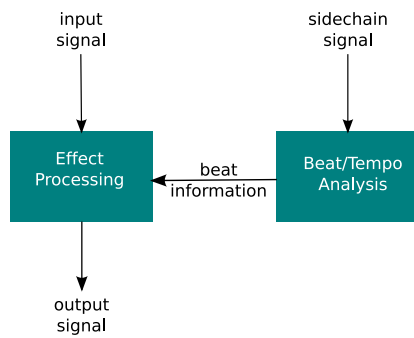
The effects were implemented in real-time as Audio Unit plug-ins and as objects for Max/MSP<sup>1</sup>. These implementations allow control of the main features of the effect through a single parameter. This parameter is the delay in beats,  $\lambda$ , for the delay effect (see Figure 3) and the number of cycles per beat,  $\Omega$  for the Tremolo, Auto-Wah and Flanger effects.

### 4.5. Audio Effect Evaluation

In informal listening tests the effects performed very well on signals with a steady tempo. However, due to the fact that the result of the application of an audio effect is subjective, it is very difficult to objectively quantify the performance of these systems. Furthermore, the performance of the effects is inherently linked to the performance of the beat-tracking system upon which they are built. However, we find that there are certain qualities of the effects that cause performance to be slightly better than might be expected.

Many beat-tracking errors are caused by the tracking of beats on the off-beat, the point half way between correct beats. The BeatDelay effect is not affected by this as

<sup>1</sup> <http://www.elec.qmul.ac.uk/digitalmusic/beatfx/>



**Figure 4.** Signal flow for Sidechaining using Beat or Tempo-Synchronous Effects

it only uses tempo information to create the effect. The effects based upon the beat-synchronous LFO can also be unaffected in certain situations. For example, if the rate of the LFO is set at an even number of cycles per beat, then the tracking on the offbeat still leads to the same processing as if the beat tracking was correct.

#### 4.6. Sidechaining using Beat and Tempo-Synchronous Effects

It is also possible to use the audio effects in a situation that uses a 3rd party signal for the beat tracking analysis and then applies the effect to the input signal. This is known as *sidechaining* and is depicted in Figure 4. The rhythmic information is extracted from the sidechain, which could be drums or other percussive instruments, and this has the advantage that the signal to which the effect is applied does not need to have a strong beat structure and can simply be a sustained note or other behaviour which might otherwise adversely affect the beat tracker.

### 5. DISCUSSION

The effects described here provide an intuitive interface to beat and tempo dependent audio effects. This makes it very simple to create a certain effect and to use it regardless of tempo. The effects have been tested predominantly with a guitar but also using an electronic piano input. The BeatDelay effect can be used to cause the notes of an arpeggio to fall upon and harmonise with the notes that follow them if the delay is set to a single beat. Variations on this effect can be created by setting the delay in beats to 2, 3, 4 or even 0.5. The BeatTremolo effect can be used to create a rhythmic pulsing effect that locks to the tempo of the input signal.

Due to the rhythmic nature of the output of these effects, the user may need to learn to ignore the effect until it synchronises correctly with the input signal. This is most common at the beginning of a musical performance where the beat tracker must be allowed to accumulate several seconds of audio to reliably infer the beats. Until this

time, the rhythmic feedback from the effect can be confusing and so the user must learn to force the effect to follow the human tempo and not be influenced by the output before correct synchronisation.

Further work on these effects may include the extraction of time signature and bar line information. This information could be used to enhance effects that use an LFO at a rate where each cycle is of a length of one or more beats by causing the start of LFO cycles to coincide with the beginning of a bar.

### 6. CONCLUSION

We have seen that beat-tracking can be effectively applied to audio effects to create effects that synchronise automatically with the beat and tempo of the input signal. These developments add a new dynamic to currently existing audio effects encouraging exploration with previously difficult to achieve effects. Furthermore the musician is free from restrictions such as set tempi and the need to update foot pedals. These effects offer the potential to greatly increase the capability of musicians to create effects that are related to the tempo of the piece in live performance situations.

### 7. REFERENCES

- [1] M. E. P. Davies and M. D. Plumbley. "Context-dependent beat tracking of musical audio". *IEEE Transactions on Audio, Speech and Language Processing*, 15(3), 1009-1020, 2007.
- [2] Hainsworth, S., *Techniques for the Automated Analysis of Musical Audio*, PhD thesis, Department of Engineering, Cambridge University, 2004.
- [3] Robertson, A. and Plumbley M.D., "B-Keeper: A beat tracker for real time synchronisation within performance". In *Proceedings of New Interfaces for Musical Expression (NIME 2007)*, New York, NY, USA, June 6-10, 2007, pp 234-237, 2007.
- [4] Stark A.M., Davies M.E.P. and Plumbley M.D., "Audio effects for real-time performance using beat tracking", In *Proc. 122nd AES Convention*, preprint 7156, Audio Eng. Soc., Vienna, 2007.
- [5] Stark A.M., Davies M.E.P. and Plumbley M.D., "Real-time beat-synchronous audio effects". In *Proceedings of New Interfaces for Musical Expression (NIME 2007)*, New York, NY, USA, June 6-10, 2007, pp 344-345, 2007.
- [6] Verfaillie, V., Zolzer, U., Arfib, D., "Adaptive digital audio effects (A-DAFx); A new class of sound transformations", *IEEE Trans. on Audio, Speech and Lang. Proc.*, Vol. 14, No. 5, pp 1817-1831, 2006.
- [7] Zolzer, U. (Ed.), Dutilleux, P., *DAFX: Digital Audio Effects*, John Wiley and Sons, 2002.