Lab Report 2

Peregrin Chiara 20027494

Digital Audio Systems, DESC 9115, Semester 1 2014
Graduate Program in Audio and Acoustics
Faculty of Architecture, Design & Planning, The University of Sydney

Abstract
The random beats processor presented here is designed to quickly generate interesting results for music composers and producers by applying audio effects to an input signal. Audio effects are pseudo-randomly assigned to rhythmically subdivided sections of an audio file. The function has been optimised to work with truncated loops where the number of beats is generally known. However, applying the process to any musical source can potentially attain interesting results. The division of the audio file into beats operates as a quasi-beat tracker and in a later version a true beat tracker is intended to be included in the processor.

Introduction
The beat subdivision of a bar is the foundation for a song’s tempo. Tempo will often be described in beats per minute. This is true for popular music styles that often have a strongly rhythmic character. In general popular music is in 4/4, or common time, although triple time metres such as 3/4 or 6/8 are also fairly commonly found. More rarely, odd time metres such as 7/4 or 5/4 are found. Regardless of the number of beats in a bar, if the time signature is known the number of beats can be specified in a function that utilises this information to process audio on a per beat basis.

Composers and music producers, especially those working in electronic genres of music and sound design, often use audio effects as integral parts of the compositional process. Having tools at their disposal to quickly generate unpredictable results can add colour and texture to composition and enhance the creative process. The random beats processor has been designed with this in mind: to provide a tool for quickly generating new and unpredictable ideas.

Implementation
In the random beat processor function, an audio input signal is first divided into a number of beats specified by the user. Typically, this would be four beats per bar for a bar of 4/4, thus for a two bar loop there would be a total of eight beats. Smaller rhythmic subdivisions can also be specified such as eighth note, sixteenth note or other rhythmic subdivision at the discretion of the user. In its current state the random beat processor requires the user to choose the number of beats or beat subdivision, however, to simplify the process a switch block with preset beat and subdivision selections could easily be added. In this way the user would only have to choose an integer value (via a potentiometer for example) between 0... n, n representing the smallest beat subdivision sought.

‘Beatlength’ is defined in the function as the input signal x divided by the specified number of beats. Depending on the sample rate this could result in beat lengths of non-integer values. Therefore ‘beatlength’ is rounded to the nearest
integer value. A Matlab SignalSource object assigns samples per frame to equal ‘beatlength’. The object prepares each beatlength for independent audio processing. Each beatlength is then processed from a selection of audio effects that are pseudo-randomly chosen using the function “randi’. The processed beatlengths are lastly concatenated to output a processed monoaural audio signal of equal length (ignoring round off errors) to the input signal. Figure 1. describes this process.

Effects processing
The audio effects presented in this implementation are for demonstration purposes to highlight the versatility of the random beat processor. Indeed, any number of effects could be included at the discretion of the user and the function modified accordingly. The Matlab implementation of a wah wah and ring modulator were sourced from Marshall (2009) and the IIR comb filter from Zölzer, ed. (2002) and adapted as functions to work within the random beat processor. This section will provide a description of each audio effect.

Ring modulation
In the digital domain ring modulation is a straightforward process of multiplication. An input signal called the modulator is multiplied by a carrier wave to attain an output signal.

\[ x(n) \cdot m(n) = y(n) \]

The carrier wave is typically a sinusoidal wave and is not audible at the output. The spectrum of the output is composed of two copies of the input spectrum called the upper side band (USB) and lower side band (LSB). The LSB is reversed in frequency and the spectrum of the input is shifted around the carrier frequency. (Zölzer (ed.), 2002, p. 76) Ring modulation creates an interesting metallic tonal effect and the three settings in a switch block of the random beat processor were chosen for the purposes of tonal variety. The choice of settings was largely based on experimentation.

Delay
The delay effect was created using an infinite impulse response comb filter. The IIR comb filter uses the difference equation and transfer function given by:

\[ y(n) = cx(n) + gy(n-M), \text{ with } M = \tau/f_s \]
\[ H(z) = c/(1 - gz^{-M}). \]
The infinite impulse response filter feeds the signal back upon itself, attenuated by a gain factor \( g \) on each cycle. The feedback loop is infinite, however, \( g \) will determine how rapidly it decays toward 0. For the stability of the system it is important that the absolute value of \( g \) is less than 1, otherwise the gain will increase infinitely. As the system can produce high amplification the input is sometimes scaled by \( c \). (Dutilleux & Zölzer in Zölzer, ed., 2002, p. 65) Figure 2 shows the signal flow of the filter. For the random beat processor, three IIR comb filters were chosen, each with different delay settings to provide variety in the processed signal. The delay settings were as close to multiples of each other with 31, 63 and 125ms delay times respectively.

![IIR Comb Filter signal flow chart. Adapted from Marshall (2009)](image)

**Wah wah**

The so-called wah-wah effect is created by time varying a bandpass filter. The wah-wah sound is characterised by a resonant centre frequency with narrow bandwidth that sweeps between a specified lower and upper frequency. The output signal is mixed with the direct signal. See figure 3. for a diagram illustrating the signal flow.

In this particular Matlab implementation of a wah-wah adapted from O’Malley, in Marshall (2009) a digital state variable bandpass filter is used. The digital state variable filter provides low, high and band pass outputs in addition to independent control over the cut-off frequency and damping factor of the filter. (Dutilleux & Zölzer in Zölzer, ed., 2002, pp. 35-36) From the input signal \( x(n) \) the three possible outputs are \((y_l)n\) for the lowpass, \((y_b)n\) for the bandpass and \((y_h)n\) for the highpass output. The difference equations for the outputs are given by:

\[
(y_l)n = (F_1)y_b(n) + y_l(n - 1) \\
(y_b)n = (F_1)y_h(n) + y_b(n - 1) \\
(y_h)n = x(n) - (y_l)(n - 1) - Q_1y_b(n - 1).
\]

The tuning coefficients \( F_1 \) and \( Q_1 \) are related to the tuning parameters \( f_c \) and \( \zeta \) as:

\[
F_1 = 2\sin(\pi f_c / f_s) \quad Q_1 = 2\zeta
\]
the low pass transfer function can be shown as:

\[ r = F_1 \quad q = 1 - F_1 Q_1 \]

\[ H(z) = \frac{r^2}{1 + (r^2 - q - 1)x^{-1} + qz^{-2}} \]

Difference equation from (Dutilleux & Zölzer in Zölzer, ed., 2002, p. 36)

This allows for independent control of the above-mentioned parameters of the input signal. For the random beat processor three settings were used again after experimentation to create movement in the output signal.

Figure 3. Wah wah signal flow chart. Adapted from Marshall (2009)

**Conclusion**

The random beat processor has been designed to meet a creative need of audio professionals including music composers and producers to create interesting variations of an audio signal by processing with audio effects in a pseudo random way. It has been presented here using three audio effects: ring modulation, delay and wah-wah, however, many other effects could be substituted or used in addition to the ones presented. The processor could potentially be used for longer input signals with a wide variety of effects to create randomised textures. It was also tested with a noise gate added prior to random processing which emphasised the rhythmic effect of a drum loop. There is potential to develop the concept further, yet even in its current state it provides an interesting compositional tool for composers, sound designers and music producers. Research into beat tracking is currently underway and it is the intention of the author that future versions of the processor will include a beat tracker with the potential for real time applications.

**Bibliography**
