**Toby Gifford**
Queensland University of Technology
Victoria Park Road
Kelvin Grove, 4059
Australia
toby.gifford@qut.edu.au

**Andrew R. Brown**
Queensland University of Technology
Victoria Park Road
Kelvin Grove, 4059
Australia
a.brown@qut.edu.au

# Stochastic Onset Detection: An approach to detecting percussive attacks in complex audio

## Abstract

*This paper introduces a new onset detection algorithm for the extraction of percussive attack times from a musical audio signal. The crux of the technique is to search for patterns of increasing noise in the signal. We therefore refer to it as the Stochastic Onset Detection (SOD) technique. This technique is designed for use with complex audio signals consisting of both pitched and percussive instrumental sounds together, and aims to report solely on the timing of percussive attacks. In contrast to most onset detection algorithms it operates in the time domain and is very efficient; suiting our requirements for real-time detection. In this paper we describe our approach to onset detection, compare this with other approaches, outline our detection algorithm and provide preliminary results from musical trials to validate the algorithm's effectiveness.*

## Introduction

The extraction of onset time information from musical signals is an important process for a number of applications. These include music information retrieval (MIR) systems seeking note and pulse identification, beat tracking systems for rhythmic segmentation, and real-time interactive music systems where music analysis and synchronisation are the goals. A number of publications have surveyed the techniques for onset detection (Bello et. al., 2004, Collins 2005). These surveys reveal that these techniques generally operate in the frequency domain and perform best on a particular class of onsets, with the most important class distinction being between pitched sounds and non-pitched sounds. The algorithm we present in this paper is targeted at onset detection in non-pitched sounds and operates in the time domain.

We have designed this algorithm for use in an interactive music system that performs real-time percussive accompaniment to a complex music signal. For example, a system that adds musical parts against an audio input, or where the system acts as an 'automatic' DJ. Existing techniques for onset detection are confounded by the presence of pitched material to varying degrees. The aim of this algorithm is to perform better than existing techniques on complex audio signals, such as recordings of multi-part performances. The results of the algorithm have been evaluated by the use of mimicry – by having the algorithm play along with the audio track triggering a percussive sound when it detects an onset. The algorithm's success was assessed aesthetically by the musicality of the output, that is to the extent it detects musically significant percussive onsets and ignores insignificant ones. Audio examples that accompany this paper can be found online at http://runtime.ci.qut.edu.au/ListeningForNoise_Examples.zip

## Existing techniques

In the survey of techniques for onset detection by Bello et. al. (2004) they describe an approach shared by many techniques; the input signal is distilled into a reduced form called the detection function; the detection function is then searched for recognisable features, often peak values; and these features are filtered, and then reported as onsets.

The simplest method for detecting onsets is to look for growth in the amplitude envelope. However, in the presence of complex audio signals containing multiple musical parts this technique is not viable.

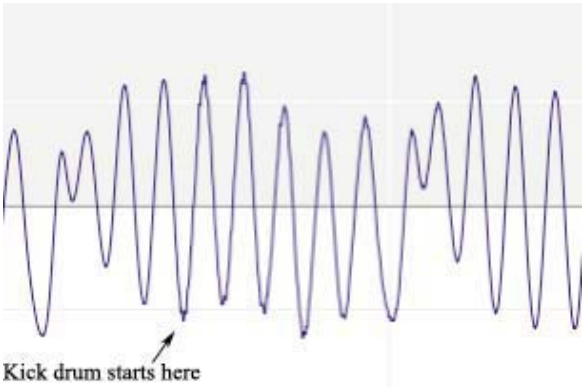


*Figure 1. Attacks can be masked in multi-part signals.*

For example, figure 1 shows the waveform for a sustained synthesizer note with a kick drum sound in the middle (corresponding to the example audio file kick_and_synth.mp3). The kick drum is clearly audible but its onset does not correspond to a peak in the amplitude envelope.

To overcome situations like this, where timbre is more significant than amplitude, a number of onset detection algorithms first split the signal into frequency bands using a Fourier transform. Onsets are then associated with growth in the energy in any band. One algorithm using this technique is Miller Puckette's (1998) bounded-Q onset detector available as the **bonk~** external for Max/MSP. Another example is the High Frequency Content (HFC) detection function of Masri and Bateman (1996) which aggregates energy across all bins but preferentially weights higher frequencies.

However, for complex audio signals in which there is power throughout the spectrum, the growth of energy in a frequency sub-band due to a percussive attack may still be masked by the ambient power of the signal in that band. The SOD technique described in this paper is designed to address this problem by seeking time domain artefacts of percussive attacks that are absent in periodic signals.

## The Rapidly Changing Component

In this paper we adopt the Deterministic Plus Stochastic model of Serra (1997) for modelling musical signals. In this model a musical signal is considered to consist of a deterministic component, which may be described as a combination of sinusoids, and a stochastic component, which is described by a random noise variable. The crux of our onset algorithm relies on the assumption that a percussive onset will be characterised by an increase in the noise component of the signal.

In Serra's model noise is equated with randomness. For example a totally random (digital) signal would be one where the amplitude of the signal at each sample point is drawn from a probability distribution and is independent of the amplitude at any other sample point.

Informally speaking our algorithm operates by separating the stochastic component from the deterministic component of the signal, and then making two queries:
(i) how loud is the stochastic component?
(ii) how random is the stochastic component?

It may seem superfluous to measure the randomness of the stochastic component; presumably if we have done a good job of the separation then it will be totally random. The reason for making this measurement is that a perfect separation is, perhaps, impossible and certainly time consuming. Instead of seeking a perfect split our algorithm operates in two steps; first by separating out a portion of the signal that contains the stochastic component plus some small amount of the deterministic component, and then estimating the fraction of this separated portion that is due to the stochastic component of the signal.

To this end we define a new notion called the Rapidly Changing Component (RCC). The RCC can be thought of as the zig-zags in the signal. For example, referring to figure 1, the signal is smooth when the synthesiser is playing on its own. The time at which the kick-drum starts is visually discernible because the signal becomes rougher (more zig-zags) at that point.

The RCC consists both of high frequency sounds and noisy sounds. Our algorithm operates by separating out the RCC from slower moving components, and then measuring the loudness of the RCC and estimating what fraction of the RCC is due to noise.

People often think of white noise as having a 'flat' Fourier spectrum, in other words equal power at all frequencies (within some band). However this picture is somewhat misleading, at least for noise as we are talking about it in this paper. In fact, if a digital signal is completely random then its spectrum is also completely random. There is a sense in which the spectrum can be described as flat - namely that the power of the spectrum in any given frequency bin will be a random number drawn from the same distribution as every other bin. But for any particular window of signal (here we are talking about the Short Time Fourier Transform with a rectangular window) the actual spectrum will not have equal power in all bins – it will be totally random. And so from one analysis window to the next there will be no relationship between the spectra, save that the total energy will be approximately the same. Consequently onset detection algorithms that operate by looking for patterns in the spectra of successive windows are ill-suited to detecting noise.

## Transient detection

A straightforward onset detection technique is to look for growth in the energy of the signal. However, for complex audio signals where the amplitude of pitched material exceeds that of the percussive onsets as shown in figure 1, simple amplitude tracking will not suffice.

As discussed above, many transient detection schemes look for growth within frequency bands. From the preceding discussion we would expect that random noise would appear in various different frequency bins inconsistently from one window to the next. This is often informally referred to as smearing[1]. One method devised to deal with this situation is the High Frequency Content (HFC) technique (Masri & Bateman, 1996). As the name suggests this approach aggregates all of the energy in high frequency bands (to be precise it aggregates all bands but linearly weights by fre-

[1] Strictly speaking smearing is where a particular frequency shows up in several frequency bins due to the quantised nature of the Short Time Fourier Transform. The use of this term in the above context is inappropriate in the same way that the use of the STFT to detect noise is inappropriate.

quency). Doing so avoids the problems of smearing to a large extent.

So why is the HFC suited to finding noise? We suggest one aspect of this can be understood with reference to Serra's Deterministic Plus Stochastic model. The stochastic component should be random at all time scales. In particular it should be random from one sample to the next, so that a burst of noise should create an increased 'jaggedness' of the signal at very short timescales, or high frequencies.

What are the drawbacks of the HFC approach? It suffers from the same basic problem as a direct amplitude approach but in more limited circumstances; if the periodic part of the signal has a lot of energy in high frequencies, then the growth in the HFC due to the percussive onset may be small in comparison to the ambient level of HFC, degrading the signal/noise ratio for the detection function.

What can be done about this? Our approach is to look at the short-timescale activity and measure how random it is. Then we can look at the growth in that randomness. This way the presence of background high frequency periodic content will not affect our detection function and the beat detection will be more robust and reliable.

## Description of the SOD Algorithm

Our Stochastic Onset Detection (SOD) algorithm is designed for real-time use with minimal latency. The input signal is processed in short windows of 128 samples. Each window we measure the level of noise in the signal. This measurement consists of four steps:

1. Separate out the RCC
2. Measure the size of the RCC
3. Measure the randomness of the RCC
4. Estimate the loudness of the stochastic component – this is our detection function.

Having obtained the most recent value for the detection function, we then employ an adaptive peak-picking algorithm (described below) to look for significant growth in the noise. Points of significant growth that exceed an absolute noise threshold are marked as percussive onsets.

## Splitting out the RCC

The first step in the construction of our noise measure is separating out the Rapidly Changing Component (RCC) from the rest of the signal. To do this we use a little rocket science - drawing inspiration from a technique developed at NASA called Empirical Mode Decomposition (Huang et. al., 1998). This is a technique for extracting 'modes' from a non-linear signal, where a mode may have a varying frequency through time. The basic idea is that to get the RCC we look at adjacent turning points of the signal (i.e., the local maxima and minima) and consider these to be short timescale activity around a carrier signal which is taken to be halfway between the turning points. It's a bit like creating a smoothed carrier wave by using a moving average with a varying order, and taking the RCC to be the residual of the signal from the carrier wave. The process is illustrated in Figure 2.

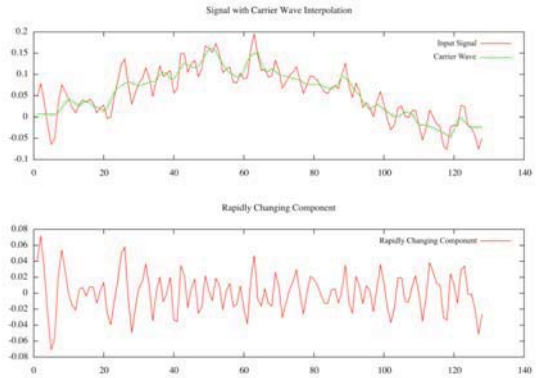


*Figure 2: Splitting out the RCC.*

## Stochastic components of the RCC

Generally the RCC will be comprised of both the stochastic component of the signal and high frequency parts of the deterministic component. So as to get a sense of the relative sizes of these contributions to the RCC, we make a measurement of the level of randomness in the RCC.

The statistic that we use to measure the level of the stochastic component is the first order autocorrelation, which measures how related the signal is to itself from one sample to the next. The stochastic component of the signal should have each sample statistically independent, and so will have an autocorrelation of zero. The deterministic component, on the other hand will be strongly related to itself from one sample to the next, and so should have autocorrelation close to one. The autocorrelation of the RCC will then reflect the relative amplitudes of these two components of the RCC; an autocorrelation of close to zero means that the RCC is mostly stochastic, whilst an autocorrelation close to one means that the RCC is mostly deterministic.

Another measure of the randomness that could be considered is the signal entropy (Shannon, 1948). The use of entropy in searching for changes in the signal noise was explored by Bercher & Vignat (2000), who give an adaptive procedure for estimating the entropy. However, their procedure is not intended for real-time use, indeed the calculation of entropy is computationally expensive (Hall & Morton, 2004). Furthermore, the autocorrelation measure has the advantage that it has a direct interpretation as approximating the percentage of the RCC that is deterministic. Conversely, if we take our measurement of randomness to be 1 – c where c is the autocorrelation, then this will be an approximate measure of the percentage of the RCC attributable to noise. For these reasons we prefer the autocorrelation measure to entropy.

## Description of Noise Measure

Having extracted the RCC we can report on how loud it is. Then, having also estimated the stochastic component of the RCC, and hence the approximate percentage of the RCC attributable to noise, we can make an estimate of the loudness of the noise in the signal by multiplying the amplitude of the RCC by it's stochastic component. In more detail, our noise measure is constructed as follows:

1. Split the signal into rectangular analysis window (we have used a window size of 128 samples).
2. Calculate the Rapidly Changing Component
   (i) Find the turning points of the signal
   (ii)The carrier wave is assumed to be halfway between adjacent turning points of the signal, so construct the carrier wave by linearly interpolating between these midpoints
   (iii) The Rapidly Changing Component is the difference between the signal and the carrier wave.
3. Calculate the size of the RCC:
   $Size_{RCC}$ = Std. Dev. of the derivative of the RCC
4. Calculate the randomness of the RCC
   $Randomness_{RCC} = 1 - autocorrelation\ of\ the\ RCC$
5. Calculate the noise
   $Noise_{RCC} = Size_{RCC} * Randomness_{RCC}$

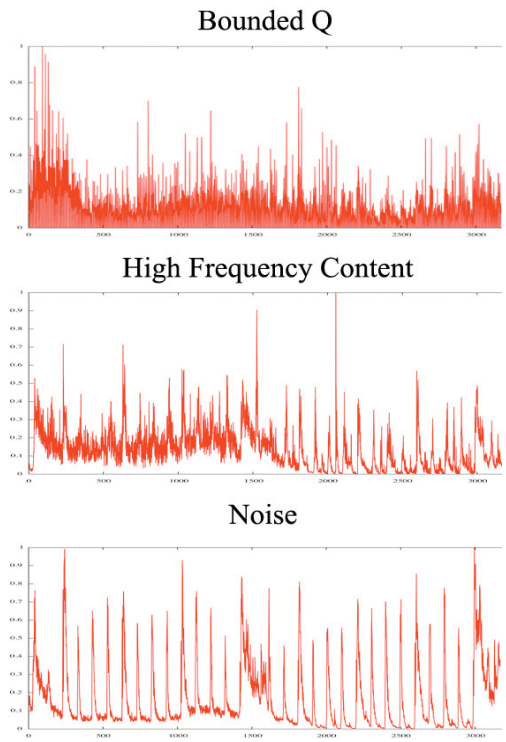


*Figure 3: Comparison of Detection Functions.*

An example of the Noise detection function for a short audio sample (corresponding to the file JungleBoogie.mp3 in the examples) is shown in figure 3. Also shown for comparison is the HFC detection function, and a Bounded Q detection function similar to that used by **bonk~**. The horizontal axis is time, and onsets are identified by looking for peaks in these detection functions. The noise detection measure has much more clearly discriminated peaks than the other two.

## Adaptive Thresholding

Having calculated the noise function we then want to identify peaks, which we will interpret as percussive attacks. In fact, what we are really looking for is sudden growth in the noise, followed by a peak, and then a decay. To do this we look for 'significant jumps' in the noise function. Different pieces of music may have markedly different noise characteristics; the size of a jump which is significant will depend on the ratio of the ambient noisiness of the pitched instruments compared to percussive instruments. To deal with this variation between musical signals we have used an adaptive thresholding technique.

We maintain a measure of the mean and standard deviation of noise in the recent past using an Exponentially Weighted Moving Average. For each new window we update these measures by accumulating a weighted value of the preceding window (we currently use a weighting of 8%). So for each new window the measures of mean and standard deviation of recent history will be 92% of what they were before + 8% of the values for the immediately preceding window. This process allows us to identify a significant jump in the noise level: where the noise level is some number of standard deviations above the mean of the recent past.

Once an onset is detected using this technique, it is not necessary to report any more onsets until the current attack is completed. A common strategy for measuring attack completion is to maintain a high and low threshold; where, for an onset to be reported the detection function must exceed the high threshold, and then no further onsets will be reported until the detection function has dropped below the low threshold. We have utilised an adaptive version of this technique for reasons mentioned previously. Once a significant jump is detected, an ongoing measure of the peak value of the detection function is maintained, and the attack is considered to be ongoing until the detection function has dropped sufficiently that recent past is significantly lower than the peak (using the same exponentially weighted moving average scheme as for detecting the onset).

The detected onsets are then further filtered by an absolute noise threshold. To be considered as an attack, a significant jump must have a peak value higher than this threshold. To allow for real-time responsiveness to the signal with minimum latency, the onset is allowed through the filter as soon as the ongoing measure of its peak value exceeds the noise threshold. For example, an open high-hat onset will have a rapid increase in the noise level but not a quick decay – so that if the algorithm were to wait until the noise had peaked

before reporting the onset it would have significant latency.

## Computational Efficiency

The stochastic onset detection (SOD) algorithm presented in this paper is quite efficient. No FFT is required because it works in the time domain. In our real-time implementation, a 128 point sample window took approximately 8 samples to process. It is also quite responsive because the RCC measurements can be calculated on small sample buffers, typically as small as 32 samples providing a latency of less than 1 millisecond.

## Experimental Results

We applied this algorithm to a selection of audio snippets containing complex audio with percussion. The snippets can be found in the online examples accompanying this paper. In addition to our few hand-selected tracks, we tested the algorithm against the MIREX Audio Tempo Extraction training data set. Training snippets in this set that did not have any percussion parts were omitted.

The Noise detection function generally seems to have a superior signal to noise ratio than the HFC or Bounded-Q detection functions. For example referring back to Figure 3, of these three detection functions the Noise detection function has the most clearly defined peaks.

We evaluated the algorithm by having it 'jam' along with the audio track (in real-time) mimicking what it hears by triggering a MIDI percussion sound when it detects an onset. The Noise measure also gives an estimate of the amplitude of the onset, and so this information is used to determine the velocity of the MIDI imitation. As a contrast, we performed the same trials using the **bonk~** external for Max/MSP.

These results are preliminary in that we have tested the algorithm with a limited range of musical examples and only performed aural analysis of the results. However, they clearly show that our approach is generally more robust than the algorithm in **bonk~** but is still not entirely consistent. In particular, our algorithm makes few mistakes in detecting onsets but does not detect all onsets. The onsets it does predict do not always correlate with those that seem most significant to human judgments, but this is not surprising given that our algorithm does not build expectations about pulse as humans do.

The algorithm appears to be particularly attuned to high-hat and cymbal onsets. For example, referring once again to Figure 3, in the snippet from Jungle Boogie, the Noise Detection algorithm follows the high-hats solidly, whilst the HFC algorithm appears more drawn to the guitar rhythm (and the Bounded-Q algorithm is totally at sea). The evaluations of these three algorithms may be heard online in the examples as JungleBoogie_nd.mp3, JungleBoogie_hf.mp3, and JungleBoogie_bq.mp3.

The other examples are of the form name_nd.mp3 for the Noise Detection sample and name_bk.mp3 for the Bonk sample.

## Conclusions

In this paper we have presented a new approach to onset detection of percussive sounds in audio signals we call Stochastic Onset Detection. This approach works with complex audio signals that have a polyphonic mixture of pitched and unpitched parts. Our approach analyses signals in the time domain and detects percussive onsets by measuring significant changes in the noise component of the signal that is typically associated with percussive attack transients. We have developed an algorithm based on this approach and provided preliminary test results that indicate that it is efficient and effective. The algorithm seems to be particularly good at detecting high pitched percussive sounds such as high-hats, which could be useful for tempo tracking of dance/rock tracks as the high-hat is often used to keep the pulse.

We hope to pursue further comparative testing with existing onset detection methods using the same hand marked test database as a benchmark for comparison used by Bello et al. (2005) and Collins (2005). We have plans to undertake future developments of this approach that include the addition of predictive assistance based on regularities and psychoacoustic models of expectation that we anticipate will particularly allow for variations in transient attack rates and allow the algorithm to have more sense of syncopated or irregular rhythms.

## References

Bello, J. P., Daudet, L., Abdallah, S. A., Duxbury, C., Davies, M., & Sandler, M. (2005). A Tutorial on Onset Detection in Music Signals. *IEEE Transactions on Speech and Audio Processing*, 13(5), 1035-1047.

Bercher, J., & Vignat, C. (2000). Estimating the Entropy of a Signal with Applications. *IEEE Transactions on Signal Processing*, 48(6), 1687-1694.

Collins, N. (2005). A Comparison of Sound Onset Detection Algorithms with Emphasis on Psychoacoustically Motivated Detection Functions. *Proceedings of the AES 118th Convention*, Barcleno, Spain.

Hall, P., & Morton, S. (2004). On the Estimation of Entropy. Annals of the *Instiute of Statistical Mathematics*, 45(1).

Huang, N., Shen, Z., Long, S., Wu, M., Shih, H., Zheng, Q., et al. (1998). The empirical mode decomposition and the Hilbert spectrum for nonlinear and nonstationary time series analysis. Proceedings of the *Royal Society of London* A, 454, 903-995.

Masri, P., & Bateman, A. (1996). Improved Modelling of Attack Transients in Music Analysis-Resynthesis. *Proceedings of the International Computer Music Conference*, Hong Kong.

Puckette, M., Apel, T., & Zicarelli, D. (1998). Real-time audio analysis tools for Pd and MSP. *Proceedings of the International Computer Music Conference*.

Serra, X. (1997). Musical Sound Modeling with Sinusoids Plus Noise. In C. Roads, A. Picialli & G. De Poli (Eds.), *Musical Signal Processing*: Swets & Zeitlinger.

Shannon, C. E. (1948). A mathematical theory of communication. *Bell Systems Tech Journal*, 27, 379-423; 623-656.