Flexible Resynthesis of Acoustic Drums

Dr Mark Sandler
Sound and Vision Engineering, Department of Electronic & Electrical Engineering, King's College London, Strand, London WC2R 2LS, UK.

Abstract
Previous papers [1-3] have described the use of Linear Predictive Coding for drum synthesis: this paper covers subsequent work performed, with particular emphasis on the signals for driving a resonant drum model. Gaussian and Brownian noise are used and these can be filtered and/or have the time envelope modified. The methods for generating these is reviewed and results presented, with some comments on their auditory performance.

Introduction

Previous papers [1-3] looked at simple Linear Predictive Analysis and Synthesis of acoustic drums. In that work, only the autocorrelation technique [4] for analysis was used and very simple stimuli were adopted: these involved short and long bursts of uniformly distributed pseudo-random noise. Another pertinent feature of that work was the extraction of the effective excitation signal, which is obtained by inverse filtering of the original sound sample by the LPC model. Although that proved an unfruitful method for driving the LPC drum models, examination of the excitation sequences themselves have led to a better synthetic driving waveform, approximately modelled by an exponentially decaying Gaussian noise sequence. Furthermore, other noise sequences with different spectral properties, in particular, Brownian noise, can also be used to give interesting synthesized drum sounds.

As well as the autocorrelation method, the covariance technique is used here, this having the advantage that the data need not be windowed: comparisons are drawn between the syntheses from the two methods. In the future other non-windowing analysis techniques, such as the Burg algorithm [5] will also be examined.

Details of LPC analysis have appeared in [1-3] and will not be repeated here, but in essence all LPC analysis techniques process a finite duration segment of a time function to produce a set of filter coefficients $\{a_i\}$. The Fourier transform of this filter is a smoothed approximation to the data's spectrum and the algorithm is formulated so that the error between the filter's spectrum and the signal's spectrum is minimized. This error can be determined by the inverse filtering operation which gives a second time sequence, which should be approximately noise-like and when applied to the filter, will give back the original data sequence.

More specifically, the LPC algorithm approximates the spectrum by the filter

$$ARp(z) = \frac{1}{1 + \sum_{i=1}^{p} a[i].z^{-i}} \qquad (1)$$

by calculating the filter coefficients $\{a_i\}$ from the time sequence $s[n]$. If this sequence has the z transform $S(z)$, then the error sequence $e[n]$ (with z transform $E(z)$) is obtained by processing $S(z)$ with the filter $A(z)$ as

$$E(z) = S(z).(1 + A(z)) \qquad (2)$$

where

$$A(z) = \sum_{i=1}^{p} a[i].z^{-i} \qquad (3)$$

For a long (and meaningful) data sequence, this procedure is applied to consecutive segments of the data: these may be overlapping, contiguous or disjoint. As applied to drums which have a slowly changing spectrum over the duration of a beat, this means that the non-stationary statistics of the signal are modelled by a sequence of stationary filter coefficient sets. For re-synthesis, the filter coefficients are changed at regular appropriate intervals.

LPC Synthesis

We now deal with the various alternatives available in regenerating drum sounds from sequences of filter coefficient sets which have been obtained by analysis. There are basically three degrees of freedom:- the type and duration of the input to the model; the type of analysis algorithm used (autocorrelation, covariance, Burg etc.); and the number of output samples computed from each filter coefficient set: these will all be discussed in turn. First though we will deal with the general principles of resynthesis. Equation (2) above can be rearranged as

$$S(z) = E(z) \frac{1}{1 + \sum_{i=1}^{p} a[i].z^{-i}} \qquad (4)$$

whose inverse z transform gives the generating expression

$$s[n] = e[n] + \sum_{i=1}^{p} a[i].s[n-i] \qquad (5)$$

so each new output sample s[n] is calculated from a new input sample e[n] and the weighted contribution of p previous outputs: the weights are provided by the filter coefficients obtained from the analysis. The basic block diagram structure of this algorithm is shown in figure 1. Every M output samples, the newest set of filter coefficients is loaded into the filter and the operation continues. Note that although the filter coefficients have changed, there is some memory of the earlier behaviour of the synthesizer in the p stored values of previous outputs and this serves to prevent discontinuities in the output sequence.

Turning now to the stimulus to the filter, e[n], it was previously stated that this might be the error sequence obtained from inverse filtering of the original data sequence. There are however a number of reasons not to use such a method. Firstly, one of the main aims of this work is to find new drum sounds, not simply to be able to exactly recreate original sampled versions - thus there is little flexibility in using the error sequence as the excitation. Second, examination of the error sequence shows that whenever the analysis moves on to a new data segment, there is a sudden increase in the energy in the error, implying that the new model does not match the signal so well at the onset of a new data sequence, and further that a slight amplitude modulation of the synthesized sound will occur at these instants. The most important feature of the error sequence is that it is noise like with a decaying exponential envelope and it is this feature that is of most use in designing excitation waveforms for use with the synthetic drum algorithms.

In the previous work, noise had been used to some effect as the input waveform. However this was simply long or short bursts of uniformly distributed pseudo-random noise as obtained from the standard random number generator available with the C compiler in use. That has now been supplemented with a high quality Gaussian pseudo-random number generator [6] which provides floating point numbers with 30 bits of precision.

Not only is this Gaussian generator used 'raw' but it also provides the input to a Brownian random number generator, which has also been used as an input signal. Brownian noise [7] is the so-called random walk signal in which each new sample is obtained as a random Gaussian displacement of the previous value. It is thus simply generated by integrating a Gaussian random number sequence, and consequently has a spectrum which rolls off as $\frac{1}{f^2}$, thereby providing strongly low-pass filtered noise.

As well as both of these primary input sequences, further flexibility is provided by an envelope function and a filter function which may be used in either order to post-process the noise sequence. The envelope generator provides both exponential and linear roll-off in the time domain, with control of the rate of roll-off and the length of the sequence. It is also possibly to delay the onset of the decay by a number of samples. The filtering operation is a standard FIR convolution algorithm, which can use filters designed with any standard FIR design algorithm, such as those available in [8].

Examples of the enveloped Gaussian noise sequence and Brownian noise sequence are given in figure 2, together with an example of an error sequence obtained by inverse filtering.

The second degree of freedom arises from the actual analysis procedure used. There is the choice of algorithm (autocorrelation, covariance, Burg and so on), the choice of model order for each data segment (all data segments have the same duration model) and the duration of each data segment, which is best expressed as the ratio of the data segment length to the model order. The more often a new model is calculated, the more closely the set of models should follow the changing spectral structure of the drum beat.

The final degree of freedom concerns the duration of each synthesized data segment (ie how many samples are recreated for each model set). This may be the same number used in the analysis, or it may be more or less. If more samples are produced per model (filter coefficient) set than were used in the analysis, the effect is to slow down the rate at which the tone of the drum beat 'bends'; if fewer samples are produced, the rate of tone bend will increase. Thus we have defined a way to produce new drum sounds from models taken from real drums.

Experiments

All the experiments were performed by simulation on a PC with math co-processor and drum samples were later replayed via a DSP card hosted in a PC, whose sole role was to provide digital to analogue conversion. The simulations were written in floating point precision in the C language. The covariance technique was used for three different drum samples and the autocorrelation method used for only one of them.

Initially three model orders were used in the analysis, 50, 100 and 200, each at three different ratios of data segment length to model order (2, 5 and 10 - approximately - viz. a x10 analysis of a 100 order model actually involves 1024 samples rather than 1000 and of a 50 order model would be 512 samples). Early resynthesis of 50 and 100 order models proved disappointing, in decaying very rapidly to zero, so that all results presented here are based on 200 order models.

For the synthesis, three basic input signals were used:- 'raw' Gaussian noise, exponentially decaying Gaussian noise and 'raw' Brownian noise. Three lengths of each type of stimulus were used, with rms values of 0.5 and 0.1 for the Gaussian signals and also for the Gaussian input to the Brownian generator. Lengths of 100 samples (≈2.25 msecs), 200 (4.5 msecs) and 500 samples (9 msecs) were used for the Gaussian signals; lengths of 500 (9 msecs), 1000 (18 msecs) and 2000 samples (36 msecs) were used for the enveloped Gaussian and Brownian signals. The enveloped signals were calculated to decay to 1% of their rms value by the end of the sequence

Three different original drum samples were used: 'x', 'y' and 'z'. All these were sampled at 20 kHz, so that this is also the re-synthesis sampling frequency. The 'y' beat is the only one analysed by both LPC algorithms and is shown in figure 3.

Results and Discussion

It is desirable to use a longish excitation sequence to stimulate the drum model, but using a long 'raw' Gaussian sequence gives a rasping onset to the beat. It is therefore often better to use an enveloped Gaussian sequence, so that energy is provided to the system over an appreciable period to continue stimulating the model. Using a sequence which decays to negligible levels after about 1000 samples (ie about 50 msecs) seems to provide good results. The longer this sequence, the more the higher frequency modes of the resonant model are stimulated. Interestingly, the response of the models obtained with an autocorrelation algorithm were different to those obtained by a covariance algorithm and they always sound much more realistic and lively, when all other variables in analysis and synthesis are kept the same.

For example, providing 200 samples of Gaussian noise to the covariance model gives a 'chock' sound, reminiscent of a rim-shot (ie very little clearly discernible resonance) whereas the skin can very definitely be heard to 'sing' when using the autocorrelation derived model. These beats are shown in figure 4 a & b. In general the autocorrelation model always needed far less energy to stimulate it and the pitch bend, characteristic of drums as the deformed skin returns to its rest position, is much better reproduced.

The same conclusion holds when the model is stimulated with enveloped Gaussian noise, though the covariance model produces a more convincing result when driving it with this longer, damped sequence. Examples of the output obtained are given in figure 4 c&d for the autocorrelation model and the covariance model respectively. The advantage of this method of stimulus is that it allows a large amount of energy to be input to the drum model, without over-driving it.

The 'x' and 'z' models, both of which were obtained only from a covariance analysis were much better than the 'y' model counterparts - example plots are not shown here for reasons of space. Clearly the issue of which analysis technique to use is not a simple one and depends on the actual original drum sample. Since autocorrelation analyses of the 'x' and 'z' models was not performed, no full conclusion can be drawn here, except to say that the pitch change of the autocorrelation 'y' models was still better than that of the covariance analyses of the other drums, so it is likely that for these drums also, the autocorrelation algorithm will provide a better model than the covariance algorithm.

Brownian noise puts significant low frequency energy and little high frequency energy into the system. Listening to this, the sound is similar to striking a drum with a soft beater, which makes intuitive sense, as a beater is unlikely to impart much high frequency energy. Examples are shown in figure 4c&f. Once again the autocorrelation and covariance derived models performed qualitatively differently; again the changing pitch over the beat's duration is clearly discernible and the whole sound is more lively with an autocorrelation model.

When longer Brownian sequences (2000 input samples) were used, a double beat effect is observed as shown and this can be connected with the slowly changing dc level of the Brownian sequence. The auditory effect of the use of Brownian noise will depend much more strongly on a particular sequence, whereas as any Gaussian sequence longer than just a few samples will be broadly the same. Thus for further flexibility in resynthesis, a number of different Brownian sequences could be available, generated from Gaussian sequences with different seeds to the generator.

Considering now the use of different amounts of synthesised samples per model set (stretching and shrinking the drum beats), see figure 4g&h for example, which show a stretched by 2 (a) and a shrunk by two version of 4d. It was much harder to hear the effect with the covariance models than was the case for the autocorrelation model, for which the pitch change is clearly evident - when shrinking the drum by producing fewer samples per model set, the pitch reduction is greater than when stretching it. It is also clear that changing the model set more rapidly keeps the resonant system 'live', the response rings on much more than the standard resynthesis. It also exhibits the complicated resonant structure of the original drum beat.

Changing the mode of stimulus to an LPC encoded drum sound gives a wide variety of sonic options. If the energy and/or duration of say an exponentially decaying

Gaussian stimulus is varied, this provides a subtle variation of sounds, consistent with a drummer striking his drum in slightly different ways. To give a degree of realism, these energy and duration parameters may be varied in some random way if a drum machine is being used or might be under user control, say by being influenced by the force or position of strike on a synthetic drum pad. The drummer may similarly simply change to effectively using soft beaters by changing the stimulus to a Brownian noise sequence.

Changing the number of samples reproduced per model set also has a subtle effect, which is most prominent when using Brownian noise as the stimulus. The difference is a slight change of pitch, as expected. The difference is best detected when a synthetic drum with double the normal number of samples per coefficient set is alternated with one reproduced with half the normal number - the latter has the lower pitch. The difference is likely to be more marked if say four or eight times the normal number (equivalently a quarter or an eighth the number) of samples are produced per coefficient set, but this was not tested.

Overall, the results were slightly disappointing in one respect, which is that the resonances were still much more damped than in the original drum samples, especially with the non-windowing covariance technique. There are some possible measures to take to alleviate this, such as performing the analysis over the same number of samples (eg over 2048 samples for a 200 order model) but overlapping the data, thereby providing more coefficient sets per drum beat and also reducing the data compression achieved.

Conclusions

Interesting sounds can be created with a wide degree of variety from a single LPC encoded drum sample which also offers significant data compression possibilities compared to sampling techniques, though obviously with the added computational burden of recreating the samples. This flexibility should be incorporated into user interfaces for synthetic drum kits.

How good the synthesised samples are depends very strongly on how the analysis was performed in the first place, with the autocorrelation algorithm standing out as the better of the two techniques. Since there are many different algorithms for obtaining LPC models, further investigation is needed into the subject of the analysis algorithm. It may prove in the long term that each algorithm has its own interesting features and that the user of the synthetic drum kit of the future will be able to experiment with these.

This paper has only really scratched the surface of the problem of synthesising drums from LPC models - there is still a great deal of experimentation that could be done. Furthermore, these same techniques could be applied to the analysis and synthesis of other instruments, perhaps starting with piano strings, which are also struck, but in the future extending to all other acoustic instruments.

References

[1]M.B.Sandler:"AR Modelling and Synthesis of Acoustic Instruments", AES Convention, March 89, Hamburg.

[2]MB Sandler:"Analysis and Synthesis of Atonal Percussion Using High Order Linear Predictive Coding", Applied Acoustics Special Issue on Musical Acoustics, May 1990.
[3]M.B.Sandler:"High Order LPC Analysis/Synthesis of Atonal Percussion", Proceedings of Institute of Acoustics,Vol. 10, part 7, p113-120.
[4]I.H.Witten:"Principles of Computer Speech", Academic Press, 1982
[5]R.A.Roberts & C.T.Mullis:"Digital Signal Processing", Addison-Wesley,1987
[6]W.G.Chambers:"Clock controlled Shift Registers in Binary Sequence Generators", IEE Proc Pt E, Vol 135, p17-24, 1988.
[7]H-O.Peitgen & D.Saupe (eds):"The Science of Fractal Images", Springer-Verlag, 1988.
[8]"Programs for Digital Signal Processing" ed by Digital Signal Processing Committee, IEEE Press, 1979.

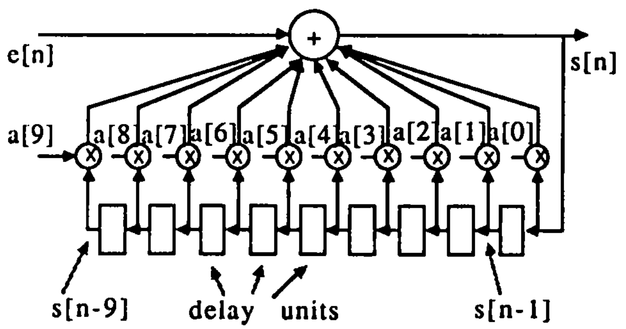


Figure 1: LPC synthesis showing filter coefficients

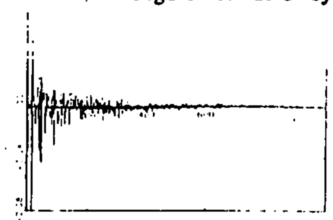


Figure 2a: Error sequence generated by inverse filtering

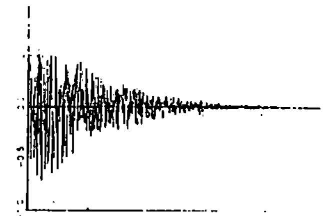


Figure 2b: Enveloped Gaussian random number sequence

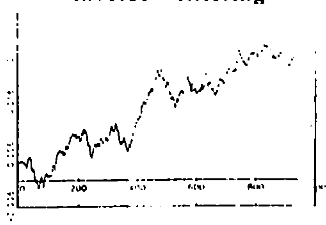


Figure 2c: Brownian random number sequence

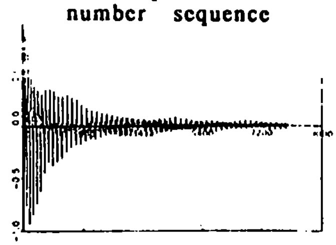
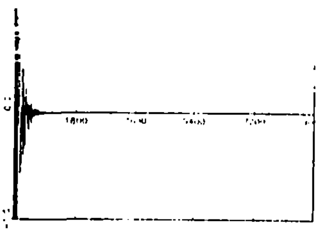
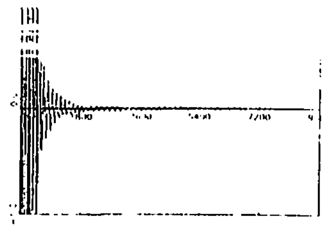


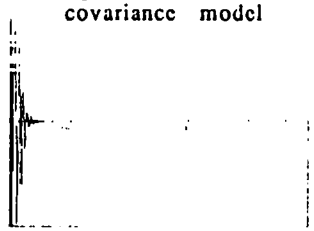
Figure 3: Original Drum 'y'.



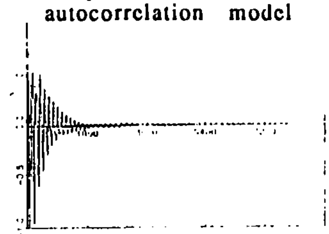
a): 100 samples of Gaussian noise into covariance model



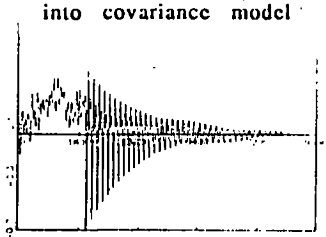
b): 100 samples of Gaussian noise into autocorrelation model



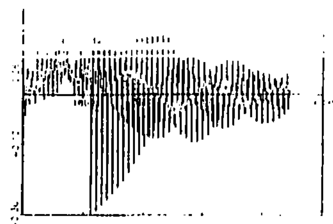
c): 500 samples of enveloped Gaussian into covariance model



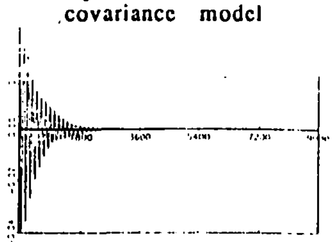
d): 500 samples of enveloped Gaussian noise into autocorrelation model



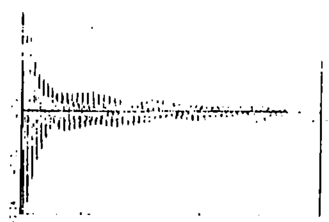
e): 2000 samples of Brownian noise into covariance model



f): 2000 samples of Brownian noise into autocorrelation model



g): 500 samples of enveloped Gaussian noise, "double" samples per coefficient set - stretched



h): 500 samples of enveloped Gaussian noise, "half" samples per coefficient set - squashed

Figure 4:Drum resynthesis with various parameters