Välimäki, Vesa; Rämö, Jussi; Esqueda Flores, Fabian

Creating endless sounds

Published in:
Proceedings of the 21th International Conference on Digital Audio Effects

Published: 04/09/2018

Document Version
Publisher's PDF, also known as Version of record

CREATING ENDLESS SOUNDS

Vesa Välimäki, Jussi Rämö, and Fabián Esqueda *

Acoustics Lab, Department of Signal Processing and Acoustics
Aalto University
Espoo, Finland
vesa.valimaki@aalto.fi

ABSTRACT

This paper proposes signal processing methods to extend a stationary part of an audio signal endlessly. A frequent occasion is that there is not enough audio material to build a synthesizer, but an example sound must be extended or modified for more variability. Filtering of a white noise signal with a filter designed based on high-order linear prediction or concatenation of the example signal can produce convincing arbitrarily long sounds, such as ambient noise or musical tones, and can be interpreted as a spectral freeze technique without looping. It is shown that the random input signal will pump energy to the narrow resonances of the filter so that lively and realistic variations in the sound are generated. For real-time implementation, this paper proposes to replace white noise with velvet noise, as this reduces the number of operations by 90% or more, with respect to standard convolution, without affecting the sound quality, or by FFT convolution, which can be simplified to the randomization of spectral phase and only taking the inverse FFT. Examples of producing endless airplane cabin noise and piano tones based on a short example recording are studied. The proposed methods lead to a new way to generate audio material for music, films, and gaming.

1. INTRODUCTION

Example-based synthesis refers to the generation of sounds similar to a certain sound but not identical. In audio, example-based synthesis solves a common problem, which we refer to as the small data problem. It is the opposite of the big data problem in which the amount of data is overwhelming and the challenge is how to find some sense of it. In the small data problem in audio processing, there may be only a few or even a single clean audio recording representing desirable sounds. It is usually unacceptable to only use that single sample in an application. For example, in various simulators, such as flight simulators [1] and working machine simulators [2], there is a need to produce a variety of sounds based on example recordings.

Previous related works have studied the synthesis of sound textures to expand the duration of example sounds. For some classes of sound, the concatenation and crossfading of samples can be quite successful. Fröjd and Horner have investigated such methods, which are related to granular synthesis [3]. They show that the method is particularly successful for the synthesis of seashore, car racing, and traffic sounds. Schwarz *et al.* compared several related approaches and showed that they perform slightly better than randomly chopping the input audio file into short segments [4]. Siddiq used a combination of granular synthesis and colored noise synthesis to produce for example the sound of running water based on modeling [5]. Both the grains and the spectrum of the background noise were extracted from a recording. Charles has also proposed a spectral freeze method, which uses a combination of spectral bins from neighboring frames to reduce the repetitive "frame effect" in the phase vocoder [6].

In this work, we use very high-order linear prediction (LP) to extract spectral information from single audio samples. The use of linear prediction has been common in audio processing for many years [7,8], but usually low or moderate prediction orders are used, such as about 10 for voice and between 10–100 for musical sounds. The use of a very high filter order is often considered overmodeling, which means that the predictive filter no longer approximates the spectral envelope, but it also models spectral details, such as single harmonics.

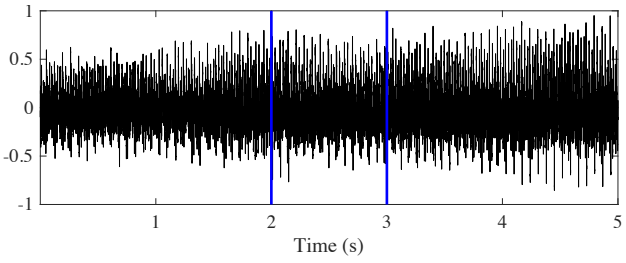
The idea and theory of utilizing higher-order LP is presented in Jackson *et al.* [9] and in Kay [10], where they studied the application of estimating the spectrum of sinusoidal signals in white noise. More recently, van Waterschoot and Moonen [11], and Giacobello *et al.* [12] have applied high-order linear predictors (order of 1024) to model the spectrum of synthetic audio signals consisting of a combination of harmonic sinusoids and white noise.

In this study we propose to use even higher orders than 1024 to obtain sufficiently accurate information, because we want to model multiple single resonances appearing in the example sounds. Obtaining high-order linear prediction filter estimates is easy in practice using Matlab, for instance. Matlab's `lpc` function uses the Levinson-Durbin recursion [13] to efficiently solve for the LP coefficients, and remarkably high prediction orders, such as 10,000 or more, are feasible. Previously, high-order linear prediction has been used for synthesis of percussive sounds [14] and for modeling of soundboard and soundbox responses of stringed musical instruments [15, 16].

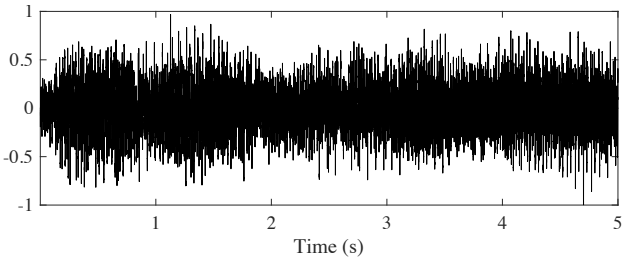
The computational cost of very high-order filtering used for synthesis is not of concern in offline generation of samples to be played back in a real-time application. However, in real-time sound generation, computational costs should be minimized. We show two ways to do so: one method replaces the white noise with velvet noise, and this leads to a simplified implementation of convolution. Another method uses the inverse FFT (fast Fourier transform) algorithm and produces a long buffer of output signal with one transformation. Neither of the methods use a high-order IIR filter, but they need its impulse response or a segment of the sound to be extended as the input signal.

This paper is organized as follows. Section 2 discusses the basic idea of analyzing a short sound example and producing a longer similar sound with life-like quality using filtered white noise. Section 3 discusses the use of velvet noise and Section 4 proposes an FFT-based method as two alternatives for the real-time imple-

* The work of Fabián Esqueda has been supported by the Aalto ELEC Doctoral School.



(a) Original waveform



(b) Synthetic waveform

Figure 1: *(a) Original airplane noise waveform and (b) a synthesized signal obtained with the LP method (P = 10,000) from the 1-second segment indicated with blue markers in (a).*

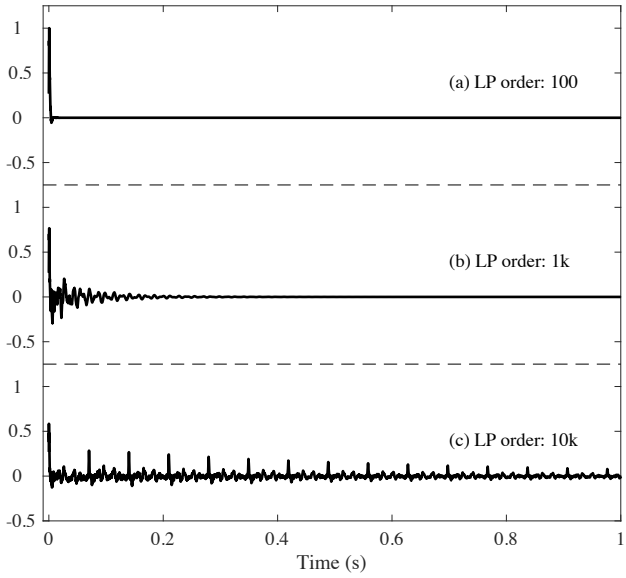


Figure 2: *Impulse responses of different order LP filters: (a) 100, (b) 1000, and (c) 10,000.*

mentation of the endless sound generator. Section 5 concludes this paper and gives ideas for further research on this topic.

2. EXTENDING STATIONARY SOUNDS

Various sounds, such as bus, road, traffic, and airplane cabin noises can be quite stationary, especially in situations where a bus is driving at a constant speed or a plane is cruising at a high altitude. Long sound samples like this are useful as background sounds in movies and games. There is also a need for sounds of this type when conducting listening tests evaluating audio samples in the presence of noise, such for evaluating headphone reproduction in heavy noise [17] or audio-on-audio interference in the presence of road noise [18].

In listening tests, controlled and stationary noises are often wanted, so that the noise signals themselves do not introduce any unwanted or unexpected results to the listening test. For example, if a short sample is looped, it may cause audible clicks each time the sample ends and restarts, or can lead to a distracting frozen-noise effect. Both irregularities can ruin a listening test. Another problem is that a recorded sample may not have a sufficiently long clean part in order to avoid looping problems. Noise recordings often include additional non-stationary audio events, such as braking/accelerating, turbulence, or noises caused by people moving, talking or coughing, which limit the length of the useful part of the sample.

These problems can be avoided by using the proposed high-order LP method. The idea is to use a short, clean stationary part of a sample (e.g. 0.5 to 1 s) to calculate an LP filter that models the frequency characteristics of the given sample. Figure 1(a) shows the waveform of a 5-second clip of airplane noise. The vertical blue lines indicate the selected clean 1-second stationary part which was used in the calculation of the LP filter.

An arbitrarily long signal can be synthesized by filtering white noise with the obtained LP filter. The resulting synthetic signal does not suffer from looping problems or include any unwanted non-stationary sound events which would degrade the quality of the signal. Figure 1(b) shows the resulting synthetic airplane noise, created by filtering 5 seconds of white noise with the LP synthesis filter calculated from the 1-second sample shown in Fig. 1(a) using prediction order of 10,000.

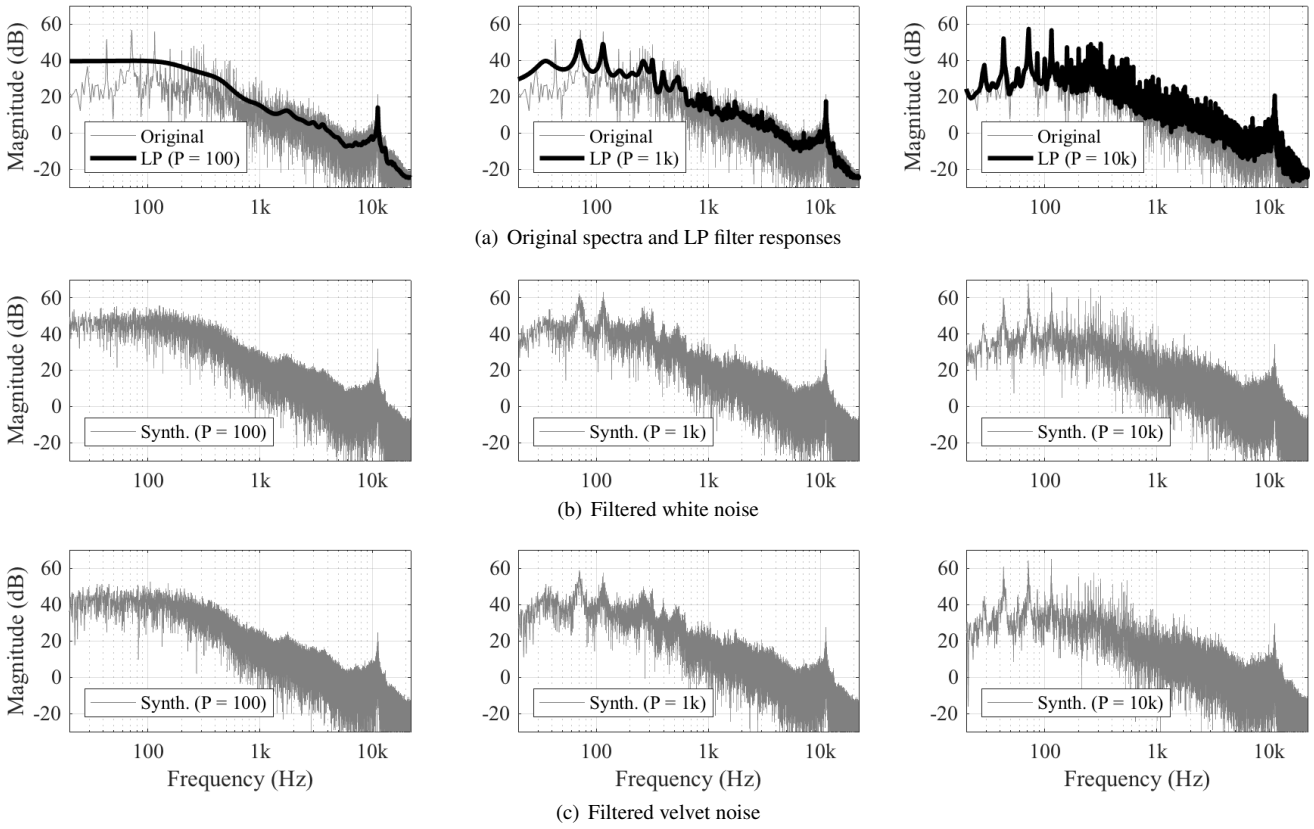
In this section, we study the synthesis of ambient noises and musical sounds using this approach. Additionally, we discuss how to change the pitch of the endless sounds.

2.1. Synthesis of Endless Stationary Audio Signals

All LP calculations in this work were done with Matlab using the built-in `lpc` function, which calculates the linear prediction filter coefficients by minimizing the prediction error in the least squares sense using the Levinson-Durbin recursion [13]. The determined FIR filter coefficients were then used as feedback coefficients to create an all-pole IIR filter, which models the spectrum of the original sample.

Figure 2 shows the calculated impulse responses of different order LP filters, where (a) is of the order of 100, (b) 1000, and (c) 10,000. As expected, the length of the impulse response increases with the LP filter order. The most interesting observation in Fig. 2 is the spiky structure of the impulse response in Fig. 2(c), where the order of the LP filter is 10,000.

Figure 3(a) shows the magnitude responses of the 1-second airplane noise sample (gray lines) from Fig. 1(a) and the magnitude response of different order (P) LP filters (black curves), i.e., from left to right the orders of 100, 1000, and 10,000 correspond to the impulse responses shown in Fig. 2. As can be seen in Fig. 3(a),



(a) Original spectra and LP filter responses

(b) Filtered white noise

(c) Filtered velvet noise

Figure 3: *Magnitude spectra of the original and synthesized airplane cabin noise. Subfigure (a) shows the magnitude spectra of an airplane cabin noise (gray lines) and magnitude responses of LP filters of different order* $P = 100, 1000,$ *and* $10,000$ *(black lines). Subfigures (b) and (c) show spectra of synthetic airplane noises created with white noise and velvet noise, respectively, using different LP filter orders.*

in order to model the low-frequency peaks of the original signal, the order P must be quite high; $P = 1000$ is not large enough to model the peak around 40 Hz, whereas $P = 10,000$ is.

Notice that in this case the order of the LP filter is very high and the filter is time-invariant, unlike in speech codecs in which the LP coefficients are updated every 20 ms or so. Thus, the whole synthesis of the sound can be conducted offline, using one large all-pole filter.

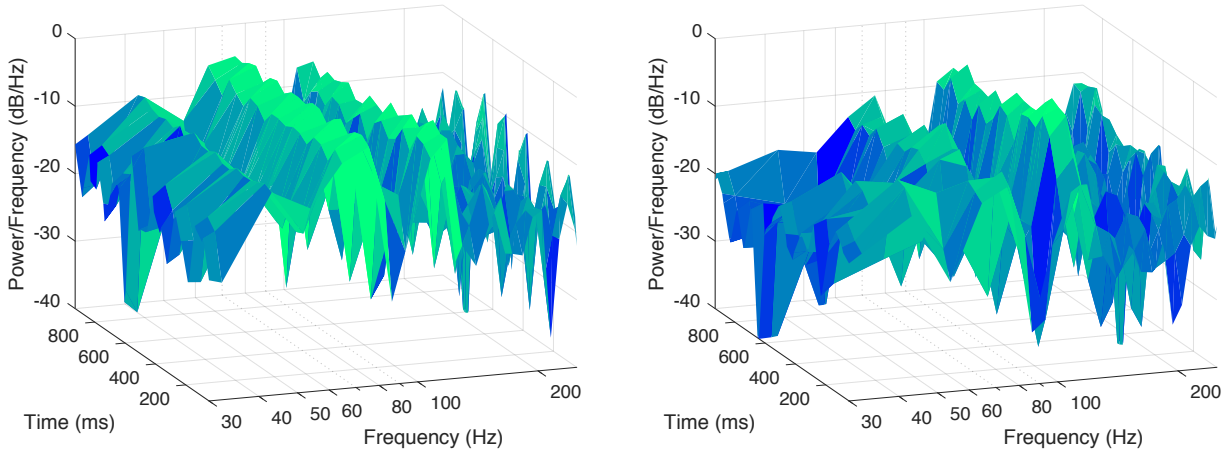
The ability of the high-order LP to capture the spectral details at low frequencies can be seen to help in the synthesis, as is shown in Fig. 3(b). In this figure, the magnitude spectra of the extended signals obtained by filtering a long white noise sequence with all-pole filters of different order are compared. It can be observed in Fig. 3(b) that using a low-order model ($P = 100$), spectral details do not appear at low and mid frequencies. However, when $P = 10,000$, the spectrum of the extended signal contains spikes even at low frequencies.

Surprisingly, although the LP filter is time-invariant, the resulting sounds are very realistic and contain lively variations. The explanation is that the white noise excites the sharp resonances of the LP filter randomly in time, making their energy fluctuate. This is illustrated in Figs. 4(a) and 4(b), which show the spectrograms of the original and synthesized airplane noise signals, respectively. As shown in the rightmost spectrogram, the signal amplitude at the resonances, excited by the white noise, is not constant and changes several dB over time. This can also be seen in Fig. 1(b), which shows the waveform of the synthesized airplane noise that is clearly fluctuating in time. In practice, the amplitude fluctuations are generally larger in the synthetic signals than in the original ones. This is not perceptually annoying, however, but rather appears to contribute to the naturalness of the extended sounds.

Furthermore, the spiky structure seen in the impulse response of the high-order LP filter, in Figure 2(c), creates natural sounding reverberance to the synthesized sound. Note that this feature is not found when the LP order is decreased to 1000, see Fig. 2(b), which otherwise sounds realistic. This implies that a fairly high LP order is required for best results.

The similarity between the magnitude response of the all-pole filter and the magnitude spectrum of the original signal suggest that it may be possible to use the original signal itself in the extension process. This idea was tested and was found to work very well: it is possible to use a short segment of the original signal, such as 0.5 s from a fairly stationary part, and use it as a filter for a white noise input. The resulting extended sound is very similar to the one obtained with high-order LP technique.

The extension technique can also be used to create tonal musical sounds using white noise as input. This has been tested with several musical signals. Figure 5 compares the spectrum of a short piano tone to that of a synthetic, extended version of the same signal. The LP filter order has been selected as 10,000 to capture the



(a) Original (b) Synthesized

Figure 4: *Spectrograms of (a) an original, and (b) LP modeled airplane noise ($P = 10,000$), from 30 Hz to 200 Hz for a 1-second sample, illustrating the fluctuation in low-frequency resonances.*

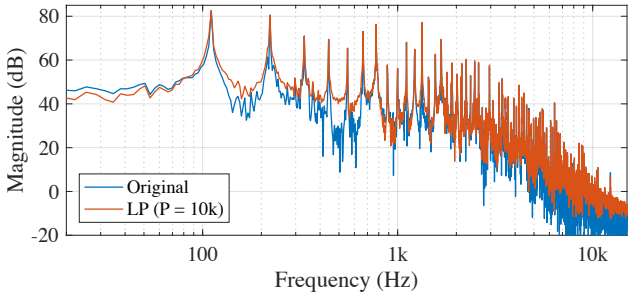


Figure 5: *Magnitude spectrum of a short piano tone (blue), and magnitude response of the LP filter (red) constructed based on that. The order P of the LP filter is 10,000.*

lowest harmonic peaks. It can be observed that the magnitude response of the filter is very similar to the spectrum of the piano tone. Listening confirms that the spectral details are preserved, and that the synthetic tone sounds similar to the original one, except that it is longer and that there are more amplitude fluctuations.

Instead of the standard LP method, it is possible to apply Prony's method or warped LP [19], for example, and hope to obtain good results with a lower model order. However, as the modeling and synthesis can be conducted offline, these options are not considered here. Instead, we will present other ideas for real-time processing in Sections 3 and 4.

The extension examples above are based on a mono signal. Pseudo-stereo signals are easily generated by repeating the extension with another white noise sequence, which is played at the other channel. This idea can be extended to more channels.

2.2. Pitch-Shifting Endless Sounds

It was found that the pitch of the extended signals can be changed easily using resampling. This is equivalent to playing the filter's impulse response at a different rate, when the output sample rate remains unchanged. A sampling-rate conversion technique can be used for this purpose.

For increasing the pitch, the sample rate of the impulse response must be lowered. Then, when the processed impulse response is convolved with white noise at the original sample rate, the pitch is increased. Similarly, the pitch of the extended sound can be lowered by increasing its sample rate and playing it back at the original rate.

This method does not require time-stretching, as the signal duration does not depend on the impulse response length. Notice that the impulse response will get shorter during downsampling and longer during upsampling, however. To better retain the original timbre, formant-preservation techniques can be used, but this topic is not discussed further in this paper.

3. REAL-TIME SYNTHESIS WITH VELVET NOISE

A direct time-domain implementation of the filtering of white noise with a very high-order all-pole filter is computationally intensive and can lead to numerical problems. It is safer for numerical reasons to evaluate the impulse response of the LP filter and convolve white noise with it. However, the computational complexity becomes even higher in this case, since there are generally more samples in the impulse response than there are LP prediction coefficients. The impulse response is often almost as long as the original signal segment to be processed. It is also possible to use the signal segment itself as the filter. To alleviate the computational burden for real-time synthesis, we suggest to use sparse white noise called velvet noise for synthesis.

Velvet noise refers to a sparse pseudo-random sequence consisting of sample values $+1$, -1, and 0 only. Usually more than 90% of the sample values are zero, however. Velvet noise has been originally proposed for artificial reverberation [20–23], where the input signal is convolved with a velvet-noise sequence. This is very efficient, because there are no multiplications, and the number of additions is greatly reduced in comparison to convolution with regular (non-sparse) white noise. Recent work also proposed the use of a short velvet-noise sequence for decorrelating audio signals [24, 25].

The convolution of an arbitrary input signal with a velvet-noise sequence can be implemented with a multitap delay line, as show in Fig. 6(a) [23]. The location and sign of each non-zero sample in the velvet noise determines one output tap in the multi-tap delay line. The sums of the signal samples at the locations of the positive and negative impulses in the velvet noise can be computed separately. Finally, the two sums are subtracted to obtain the output sample.

In the endless sound application considered in this paper, the role of the velvet noise is different than in the reverb or decorrelation application. Now, the velvet noise becomes the input signal, which is convolved with the short signal segment. The signal segment $x(n)$ can be stored in a buffer (table), and the taps of a multitap delay line, where the tap locations are determined by the velvet-noise sequence, move along it. This is illustrated in Fig. 6(b), which shows a time-varying multi-tap delay line in which the taps (read pointers) march one sample to the right at every sampling step. In this case, velvet noise can be generated in real time: every time a new velvet-noise frame begins, two random numbers are needed to determine the location and sign of the new tap. The oldest tap that reaches the end of the delay line is decimated. The computational efficiency of the proposed filtering of the velvet noise sequence is very high, as it is comparable to that of the standard velvet-noise convolution.

A velvet-noise signal with a density of 4410 samples per second (i.e., one non-zero impulse in a range of 10 samples) was used for testing this method. This corresponds to a 90% reduction in operations. Since velvet-noise convolution does not require multiplications but only additions, a total reduction of 95% is obtained w.r.t. standard convolution with white noise. In practice, the required velvet-noise density depends on the signal type. It is known that a lower density can sound smooth when the velvet noise is lowpass-filtered [20], which in this case corresponds to an input signal of lowpass type.

Figure 3(c) shows the magnitude spectra of extended signals obtained by filtering velvet noise, as described above. Comparison with Fig. 3(b) reveals that the results are very similar to those obtained by filtering regular white noise, which requires about 20 times more operations. The endless sound synthesis based on velvet-noise filtering can be executed very efficiently in real time, and additional processing, such as gain control or filtering, can be adjusted continuously. Below we propose another efficient method, which is based on FFT techniques.

4. ENDLESS SOUND SYNTHESIS USING INVERSE FFT

We propose yet another interesting technique for creating virtually endless sounds, which utilizes the concept of fast convolution [22, 26–28]. It is well known that frequency-domain convolution using the FFT becomes more efficient than the time-domain convolution when the convolved sequences are long. When two sequences of length N are convolved, the direct time-domain convolution takes approximately N^2 multiplications and additions whereas the FFT takes the order of $N \log(N)$ operations only [22, 29]. The difference in computational cost between these two implementations becomes significant even at fair FFT lengths, such as a few thousand samples.

The main point in the fast convolution is to utilize the convolution theorem [28, Ch. 11], which states that the time-domain convolution of two signals is equivalent to the point-wise multiplication of their spectra:

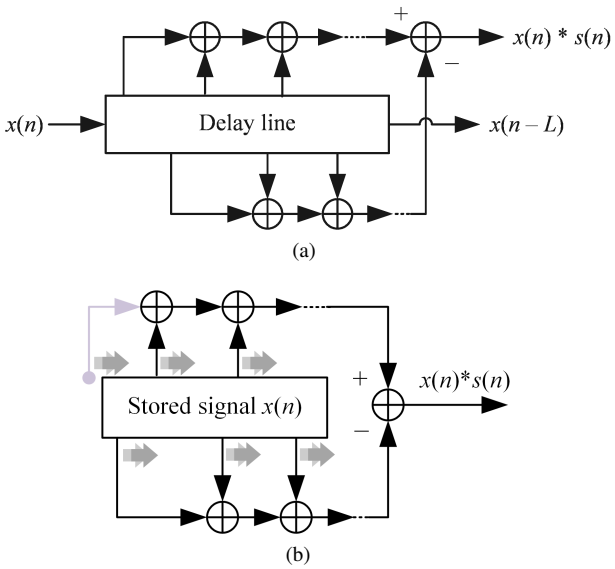


Figure 6: *(a) Convolution of an arbitrary signal $x(n)$ with a velvet-noise sequence $s(n)$ corresponds to a multi-tap delay line from which the output is obtained as the difference of two subsums. (b) Convolution of a short signal segment $x(n)$ with a velvet-noise signal can be implemented as a multi-tap delay line with moving output taps.*

$$v(n) * x(w) \leftrightarrow V(f)X(f), \tag{1}$$

where, in this application, $v(n)$ is a white noise signal and $x(n)$ is the signal segment (or the impulse response of the LP filter), and $X(f)$ and $V(f)$ are their Fourier transforms, respectively. Figure 7(a) shows a block diagram of the basic fast convolution method. Notice that the output is obtained by using the inverse FFT (IFFT).

The frequency-domain signals X and V can be written as

$$X = R_x e^{j\theta_x}, \tag{2}$$
$$V = R_v e^{j\theta_v}, \tag{3}$$

where R and θ are the magnitude and phase vectors of the two signals, respectively. Further, the multiplication of the frequency-domain signals can be written as

$$Y = VX = R_v e^{j\theta_v} R_x e^{j\theta_x} = R_v R_x e^{j(\theta_v + \theta_x)}. \tag{4}$$

By taking the IFFT of Y, one frame (N samples) of the convolved time-domain signal $y(n)$ is synthesized. As our aim is essentially to create a synthesized sound similar to the original but longer, we can apply zero padding to the short original sample, before taking the FFT, and use a white noise sequence of the same length.

Additionally, as it is known that the white noise has ideally a constant power spectrum and a random phase, the white noise can be produced directly in the frequency domain (instead of first creating it in the time domain and then transforming it to the frequency domain with the FFT). It is helpful to assume that the magnitude response of the short white noise sequence is flat, although this is not exactly true for short random signals. Siddiq used a

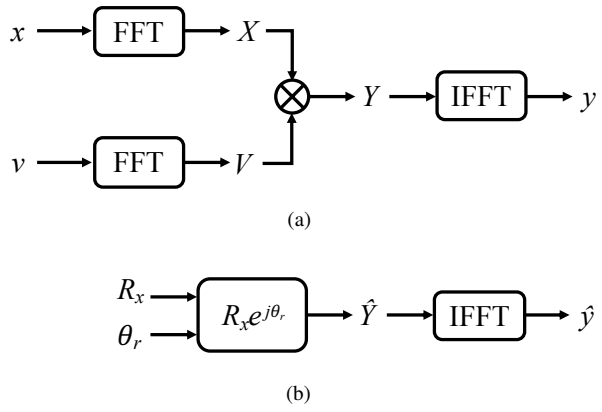


Figure 7: *(a) Regular fast convolution and (b) the proposed IFFT-based synthesis, where x is the signal segment to be extended, v is a white noise sequence, R_x is the magnitude of spectrum X, and θ_r is a randomized phase with values between $-\pi$ and π.*

similar approach to generate colored noise in granular texture synthesis [5].

Now, when we look at the last product in Equation (4), we can set the magnitude spectrum of the white noise to unity, so that the magnitude response R_x is left unchanged. Furthermore, as adding a random component to the original phase randomizes it, we may as well delete the original phase and replace it with a random one, resulting in

$$R_x e^{j(\theta_r + \theta_x)} \rightarrow R_x e^{j\theta_r}, \tag{5}$$

where θ_r is the randomized phase. Thus, the whole process of frequency-domain convolution is reduced to taking the FFT of the original signal segment (or impulse response), replacing its phase with random numbers while keeping the original magnitude, and taking the IFFT, as shown in Figure 7(b).

Stricktly speaking, in Figure 7(b) the polar coordinate inputs R_x and θ_r are transformed to Cartesian coordinates to construct $\hat{Y}$, an approximation of Y. By taking the IFFT, one frame of the time-domain waveform $\hat{y}(n)$ is obtained. Both signals R_x and θ_r can be constructed offline, R_x is the magnitude of the original sample, and θ_r is constructed as

$$\theta_r = [0,\ r,\ 0,\ -\tilde{r}], \tag{6}$$

where the two zeros in the phase vector are located at the DC and the Nyquist frequency, r contains uniformly distributed random values between $-\pi$ and π, and $\tilde{r}$ is r with reversed elements. Notice that the sign of phase values $\tilde{r}$ must be opposite to those or r, because they represent the negative frequencies. The length of both r and $\tilde{r}$ is $(N/2) - 1$, where N is the FFT length. Parameter N is chosen to be the same as the length of the zero-padded signal.

Note that with the technique described above and in Figure 7(b), R_x can be calculated directly as the FFT magnitude of the original signal, without the need of LP estimation. In fact, a high-order LP filter very closely imitates the magnitude spectrum of the signal segment. Figure 8(b) gives an example in which the same 1-second segment as in Fig. 1(a) has been employed. As can be seen, the produced signal fluctuates in a similar way as the one generated using filtering white noise with the all-pole filter.

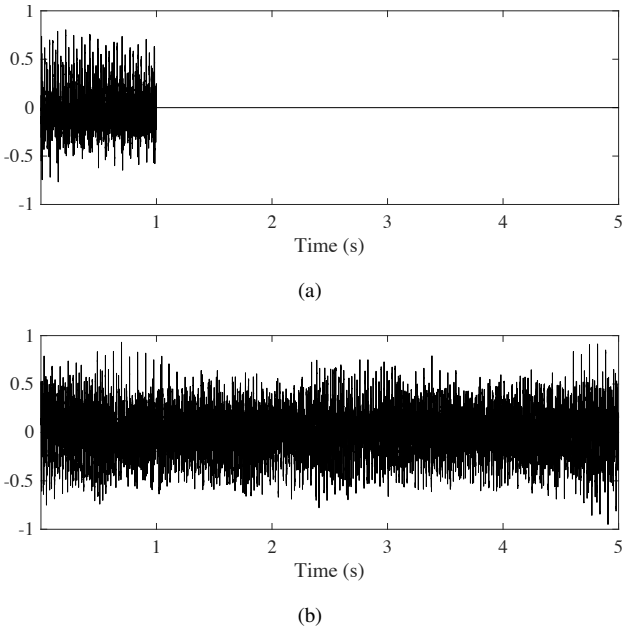


Figure 8: *(a) Original airplane noise segment (cf. Fig. 1(a)), which has been expanded with zero padding to a desired length. (b) Synthesized waveform obtained with the IFFT technique of Fig. 7(b).*

4.1. Concatenation Employing Circular Time

It is a remarkable fact that windowing or the overlap-add method are not necessary with the proposed IFFT synthesis technique. With this approach, copies of a long segment of the produced random-phase signal can simply be concatenated without introducing discontinuities at the junction points. This is a consequence of the fast convolution operation, where the time-domain representation is circular, and is therefore also called circular convolution [13, 27].

When the extended segment is long enough, such as 4 seconds or longer, it will be difficult to notice that it repeats [1]. The best option for endless sound synthesis thus appears to be to synthesize one long extended signal segment using the IFFT and then repeat it. However, if more than one extended segment is synthesized from the same input signal and they are concatenated, hoping to produce extra variation, they will usually produce clicks at the connection points. In this case a crossfade method would be needed to suppress the clicks. Naturally, this idea is not recommended, as it is much easier to produce only a single segment using IFFT and repeat it.

The next example illustrates the fact that the repetition of a single segment works fine. We use a 4000-sample segment of a piano tone as the input signal and apply the method of Fig. 7(b). The IFFT length N is 4096. Figure 9(a) shows two concatenated copies of this extended signal, leading to a signal of length 8192. Figure 9(b) zooms to the joint of the two copies, showing that there is no discontinuity, but that the end of the segment fits perfectly to its beginning.

[1] However, it has been shown in laboratory experiments that people can notice much longer repetitions in sound [30].

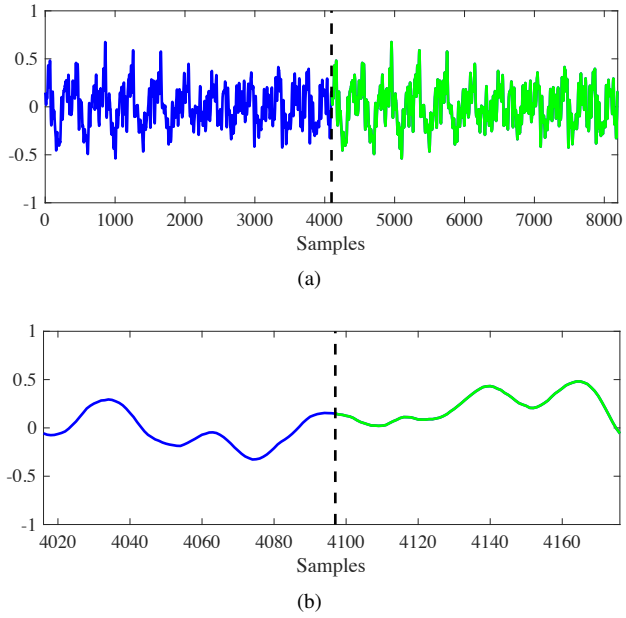


Figure 9: *Two concatenated copies of the same signal obtained with the proposed IFFT method, first copy plotted with blue line and the second with green line. Subfigure (a) shows the signals in their full length, and (b) zooms to the point where the signals are joined, illustrating the perfect fit of the junction point. The dashed vertical line indicates the beginning of the second copy of the signal.*

4.2. Comparison of Methods

So far there are three principally different methods for creating endless sounds: filtering of white noise with the LP-based all-pole filter, filtering a signal segment with velvet noise, and IFFT synthesis based on a signal segment. The filtering of regular white noise is the basic method, which also leads to the largest computational load, whereas the IFFT method is the most efficient one. Also the method based on filtering velvet noise is computationally efficient, and as it produces the output signal one sample at a time, it allows amplitude modulation or other modifications to be executed during synthesis. The filtering methods are suitable for low-latency application whereas the IFFT method is only suitable for synthesizing the signal in advance.

As a test case, we measured the time it takes to produce 1 minute of sound from a short signal segment using Matlab. For the first method, an LP filter of order 1000 was used, which produced an impulse response that could be truncated to the length 10,000 samples. The convolution of this filter impulse response with 2,646,000 samples ($60 \times 44{,}100$) of white noise took in average about 3.4 s. This is much less than 1 minute, so it should be easy to run the synthesis in real time.

For comparison, the IFFT of the length 2,646,000 produced the 1-minute segment of the extended signal at one go, and it took in average 0.14 s to compute[2]. Remarkably, practically the same result was obtained by producing 4.0 s of the extended signal with the IFFT in just about 0.0005 s, and by repeating it 15 times (at no extra cost!). As listeners do not generally notice the repetition over several seconds and as there are no clicks at the connection points, this produces equally good results as the longer IFFT synthesis.

[2]Matlab's FFT algorithm is fastest when the length is a power of 2, but 2,646,000 is not.

5. CONCLUSION AND FUTURE WORK

This paper has discussed the use of linear prediction and the inverse FFT for solving the small data problem in sampling synthesis. Useful methods were proposed to extend the duration of short example sounds to an arbitrary length. The first method employs high-order linear prediction to a selected short segment in the original recording. Surprisingly, the impulse response of the filter can be replaced with a short segment of the original sound signal.

A synthetic sound of arbitrary length may then be produced by filtering white noise with a segment of the original sound. Lively variations appear in the produced sound, as the random signal pumps energy to the narrow resonances contained in the signal's spectrum. These variations are shown to be generally larger in terms of amplitude variance than in the original sound, but they help to make the extended sound appear natural and non-frozen. Sound synthesis can take place offline so that during presentation the generated signal is played back from computer memory, like in sampling synthesis. In this case, the computational cost of running a large all-pole filter or long convolution is of no concern.

Alternatively, we proposed to reduce the computational cost for real-time synthesis by replacing the white noise signal with velvet noise or by generating the noisy extended signal using the inverse FFT from the original magnitude and a random phase spectrum. The IFFT-based method produces a long segment of the output signal at one time. Another unexpected result is that the segment produced by the IFFT method can be repeated by concatenating copies of itself without the need of windowing or crossfading. This property comes from the fact that the fast convolution, which is the basis of the proposed IFFT synthesis method, implements a circular convolution in the time domain.

Future work may consider the analysis of perceived differences in extended samples in comparison to the original recording. It would be desirable to find a method to control the fluctuations of resonances in the synthetic signal, although they are not annoying generally. It would also be of interest to consider formant-preserving pitch-shifting techniques, which could be used to build a sampling synthesizer based on the ideas proposed in this paper.

Audio examples related to this paper are available online at http://research.spa.aalto.fi/publications/papers/dafx18-endless/. The examples include synthetic signals obtained with different LP orders and IFFT lengths, and various sound types, such as the airplane cabin noise, the piano tone, a distorted guitar, and an excerpt taken from a recording by the Beatles.

6. REFERENCES

[1] H. Ploner-Bernard, A. Sontacchi, G. Lichtenegger, and S. Vössner, "Sound-system design for a professional full-flight simulator," in *Proc. Int. Conf. Digital Audio Effects (DAFx-05)*, Madrid, Spain, Sept. 2005, pp. 36–41.

[2] V. Mäntyniemi, R. Mignot, and V. Välimäki, "REMES final report," Tech. Rep., Science+Technology 16/2014, Aalto University, Helsinki, Finland, 2014, Available at https://aaltodoc.aalto.fi/handle/123456789/14705.

[3] M. Fröjd and A. Horner, "Sound texture synthesis using an overlap-add/granular synthesis approach," *J. Audio Eng. Soc.*, vol. 57, no. 1/2, pp. 29–37, Jan./Feb. 2009.

[4] D. Schwarz, A. Roebel, C. Yeh, and A. LaBurthe, "Concatenative sound texture synthesis methods and evaluation," in *Proc. 19th Int. Conf. Digital Audio Effects (DAFx-16)*, Brno, Czech Republic, Sept. 2016, pp. 217–224.

[5] S. Siddiq, "Morphing granular sounds," in *Proc. 18th Int. Conf. Digital Audio Effects (DAFx-15)*, Trondheim, Norway, Nov./Dec. 2015, pp. 4–11.

[6] J.-F. Charles, "A tutorial on spectral sound processing using Max/MSP and Jitter," *Computer Music J.*, vol. 32, no. 3, pp. 87–102, 2008.

[7] J. A. Moorer, "The use of linear prediction of speech in computer music applications," *J. Audio Eng. Soc.*, vol. 27, no. 3, pp. 134–140, Mar. 1979.

[8] P. R. Cook, *Real Sound Synthesis for Interactive Applications*, AK Peeters, Ltd., 2002.

[9] L. B. Jackson, D. W. Tufts, F. K. Soong, and R. M. Rao, "Frequency estimation by linear prediction," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Tulsa, OK, USA, Apr. 1978, pp. 352–356.

[10] S. M. Kay, "The effects of noise in the autoregressive spectral estimator," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 27, no. 5, pp. 478–485, Oct. 1979.

[11] T. van Waterschoot and M. Moonen, "Comparison of linear prediction models for audio signals," *EURASIP J. Audio, Speech, Music Process.*, vol. 2008, pp. 1–24, Dec. 2008.

[12] D. Giacobello, T. van Waterschoot, M. G. Christensen, S. H. Jensen, and M. Moonen, "High-order sparse linear predictors for audio processing," in *Proc. 18th European Signal Process. Conf.*, Aalborg, Denmark, Aug. 2010, pp. 234–238.

[13] L. B. Jackson, *Digital Filters and Signal Processing*, Kluwer, Boston, MA, USA, second edition, 1989.

[14] M. Sandler, "Analysis and synthesis of atonal percussion using high order linear predictive coding," *Appl. Acoust.*, vol. 30, no. 2–3, pp. 247–264, 1990.

[15] M. Karjalainen and J. O. Smith, "Body modeling techniques for string instrument synthesis," in *Proc. Int. Computer Music Conf.*, Hong Kong, Aug. 1996, pp. 232–239.

[16] F. v. Türckheim, T. Smit, and R. Mores, "String instrument body modeling using FIR filter design and autoregressive parameter estimation," in *Proc. Int. Conf. Digital Audio Effects (DAFx-10)*, Graz, Austria, Sept. 2010.

[17] J. Rämö and V. Välimäki, "Signal processing framework for virtual headphone listening tests in a noisy environment," in *Proc. Audio Eng. Soc. 132nd Conv.*, Budapest, Hungary, Apr. 2012.

[18] J. Francombe, R. Mason, M. Dewhirst, and S. Bech, "Elicitation of attributes for the evaluation of audio-on-audio interference," *J. Acoust. Soc. Am.*, vol. 136, no. 5, pp. 2630–2641, Nov. 2014.

[19] A. Härmä, M. Karjalainen, L. Savioja, V. Välimäki, U. K. Laine, and J. Huopaniemi, "Frequency-warped signal processing for audio applications," *J. Audio Eng. Soc.*, vol. 48, no. 11, pp. 1011–1031, Nov. 2000.

[20] M. Karjalainen and H. Järveläinen, "Reverberation modeling using velvet noise," in *Proc. Audio Eng. Soc. 30th Int. Conf. Intelligent Audio Environments*, Saariselkä, Finland, Mar. 2007.

[21] K.-S. Lee, J. S. Abel, V. Välimäki, T. Stilson, and D. P. Berners, "The switched convolution reverberator," *J. Audio Eng. Soc.*, vol. 60, no. 4, pp. 227–236, Apr. 2012.

[22] V. Välimäki J. D. Parker, L. Savioja, J. O. Smith, and J. S. Abel, "Fifty years of artificial reverberation," *IEEE Trans. Audio, Speech, and Lang. Processing*, vol. 20, no. 5, pp. 1421–1448, Jul. 2012.

[23] V. Välimäki, B. Holm-Rasmussen, B. Alary, and H.-M. Lehtonen, "Late reverberation synthesis using filtered velvet noise," *Appl. Sci.*, vol. 7, no. 483, May 2017.

[24] B. Alary, A. Politis, and V. Välimäki, "Velvet-noise decorrelator," in *Proc. Int. Conf. Digital Audio Effects (DAFx-17)*, Edinburgh, UK, Sept. 2017, pp. 405–411.

[25] S. J. Schlecht, B. Alary, V. Välimäki, and E. A. P. Habets, "Optimized velvet-noise decorrelator," in *Proc. Int. Conf. Digital Audio Effects (DAFx-18)*, Aveiro, Portugal, Sept. 2018, elsewhere in these proceedings.

[26] G. Stockham, Jr., "High speed convolution and correlation," in *Proc. Spring Joint Comput. Conf.*, Boston, MA, USA, Apr. 1966, pp. 229–233.

[27] D. Arfib, F. Keiler, U. Zölzer, V. Verfaille, and J. Bonada, "Time-frequency processing," in *DAFX: Digital Audio Effects, Second Edition*, U. Zölzer, Ed., pp. 219–278. Wiley, 2011.

[28] J. D. Reiss and A.P. McPherson, *Audio Effects: Theory, Implementation and Application*, CRC Press, Taylor & Francis Group, Boca Raton, FL, USA, 2015.

[29] J. O. Smith, *Spectral Audio Signal Processing*, Online book, http://ccrma.stanford.edu/~jos/sasp/, 2011 edition, Accessed 23 March, 2018.

[30] R. M. Warren, J. A. Bashford, J. M. Cooley, and B. S. Brubaker, "Detection of acoustic repetition for very long stochastic patterns," *Perception & Psychophysics*, vol. 63, no. 1, pp. 175–182, Jan 2001.