A New Hybrid Reverb Algorithm

Martin Vicanek

10. June 2020

1 Introduction

Artificial reverberation is an essential audio processing technique with widespread use. From a technology perspective the field is rather mature [1, 2, 3], and a huge range of products are available. Since the pioneering work of [4], today's devices operate in the digital domain and fall into one of the following two categories:

- Convolution reverbs
- Algorithmic reverbs

Convolution reverbs are considered as most naturally sounding, however they tend to be demanding on resources, and their flexibility is somewhat limited. Algorithmic reverbs, on the other hand, are often much leaner, however they may sound metallic and are difficult to tune.

In this article I present a hybrid scheme with the objective to achieve the best of both worlds.

2 Convolution

Let x_n with $n = 0, 1, 2, \ldots$ denote a stream of incoming audio samples. Reverberation may be viewed as a dense set of echoes, i.e repetitions of the input at later times and with decreasing intensity. Mathematically, this may be expressed as a convolution. A simple model uses white noise r_n and an exponential decay a^n with $|a| < 1$ as an impulse response (IR). The result is

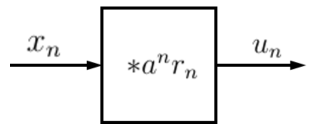

Figure 1: Convolution block diagram

a transformed stream of samples u_n,

$$u_n = \sum_{m=0}^{N-1} a^m r_m x_{n-m} \tag{1}$$

Figure 1 depicts the contents of eq.(1) in a block diagram.

White noise may not represent the particular reverberation characteristic of a *specific* space, however it has a number of desirable features which make it a preferred choice for a generic, pleasant sounding reverberation engine:

- white noise is colorless (no metallic resonances)
- white noise is structure-less (no apparent pattern in the time domain)
- white noise may be generated easily on the fly (no need to store or load large amounts of data)

The first two bullet points are features not easily obtained with algorithmic reverbs, whereas the third point is an advantage over genuine convolution reverbs.

It is common to measure the reverberation time RT_{60} as the duration for a decay to a -60 dB level. We can express a in terms of RT_{60} and the sampling rate f_s,

$$a = 10^{\frac{-3}{f_s \mathrm{RT}_{60}}}. \tag{2}$$

Figure 2 illustrates the result of the convolution operation in eq.(1) when applied to a single incoming impulse. The abrupt truncation at N samples indicates that the IR needs to be much longer for a natural reverberation tail. Indeed, convolution reverbs may process IRs with close to a million samples.

Unfortunately, such long IRs bind resources in terms of CPU demand, storage space and loading time. In view of the simple structure of the late reverberation stage, characterized by a few parameters at most, convolution modeling with some 10^5 parameters appears overkill.

Figure 2: Impulse response of a convolution with exponentially decaying white noise.

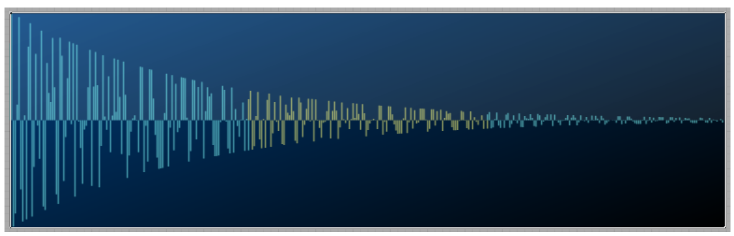

Figure 3: Impulse response of a convolution with exponentially decaying white noise followed by a comb filter. Coloring has been applied for illustration.

3 Copy & Paste

If we add suitably delayed and attenuated copies of the impulse response in figure 2 we obtain a smooth exponential tail, as shown in figure 3.

We may generate such copies recursively with a delay matching the IR size N and a feedback coefficient a^N,

$$v_n = u_n + a^N v_{n-N} \tag{3}$$

Figure 4 shows the corresponding block diagram.

Although the procedure laid out so far produces a seamlessly decaying exponential series of structure-less echo blocks, the human ear clearly detects a repeating pattern. This unwanted result may be mitigated with little effort, as described in the next section.

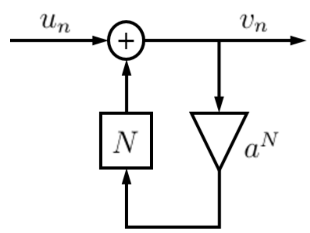

Figure 4: Comb filter block diagram. The box with an N inside represents a delay by N samples.

4 Concealing Periodicity

To avoid repetitions we need to make sure that each echo block is different. However, if we just switch to another IR every N samples, the result will no longer be seamless. One possible remedy is to split the incoming signal x_n into two separate convolution streams x'_n and x''_n with alternating weights w'_n and w''_n,

$$x'_n = w'_n x_n, \quad x''_n = w''_n x_n. \tag{4}$$

The two processing branches thus become

$$u'_n = \sum_{m=0}^{N-1} a^m r'_m x'_{n-m}, \quad u''_n = \sum_{m=0}^{N-1} a^m r''_m x''_{n-m} \tag{5}$$

IR switching is carried out when the corresponding weight is zero: At $n = 0, N, 2N, \ldots$ the IR r'_n is replaced by another, uncorrelated random sequence, and likewise at $n = N/2, 3N/2, \ldots$ for the r''_n IR. If r'_n and r''_n are uncorrelated, the weights must satisfy $w'^2_n + w''^2_n = 1$ for a perfect recombination $u_n = u'_n + u''_n$. Suitable weight functions are

$$w'_n = \sin(2\pi n/N), \quad w''_n = \cos(2\pi n/N). \tag{6}$$

It may seem strange to allow the weights to become negative, however the sign is immaterial as it may be absorbed in the random sequences r'_n and r''_n.

Figure 5 shows the corresponding block diagram.

5 Implementation Technicalities

The only free parameter in the presented scheme is the IR size N. It should be sufficiently large so modulation from cross-fading according to figure 5 does

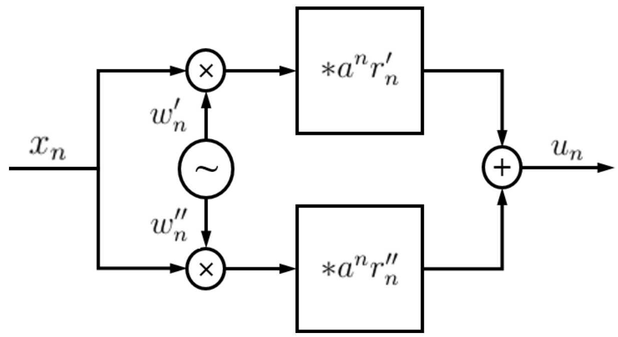

Figure 5: Crossfading convolution block diagram

not become apparent. On the other hand, increasing N results in increased CPU load. A reasonable choice was found to be $N = 8192$ at a sampling rate $f_s = 44100\ \mathrm{s}^{-1}$.

5.1 Partitioned Convolution

Convolution is efficiently carried out using the fast Fourier transform. However, a straight-forward application would introduce a latency of the IR size N, which may be prohibitive in a live context. This problem may be solved using partitioned convolution [5]. Although this method allows to remove latency altogether, it is sufficient and even desirable to leave some delay between the original signal and the reverberation onset. Good results were achieved with 8 partitions of 1024 samples each, resulting in a 1024 samples or 23 ms latency, which is appropriate for short reverberation times $\mathrm{RT}_{60} \lesssim 200$ ms. For longer reverberation times, an extra delay may be added as a predelay.

5.2 White Noise Generation

White noise for the IRs may be produced easily with a linear congruential random number generator [6]. Uncorrelated sequences are obtained by using different seeds.

Usually the processed (wet) signal will be added to a certain amount to the original (dry) signal. In order to maintain consistent effect levels for short and long reverb times RT_{60}, respectively, the white noise amplitude should be scaled with a factor $1/\sqrt{f_s \mathrm{RT}_{60}}$.

5.3 Stereo Processing

A stereo reverberation effect is obtained by applying the scheme for the left and the right channels separately with uncorrelated IR sequences for maximum stereo width. Stereo width may be reduced if necessary by intermixing left and right channels.

5.4 High Frequency Damping

In real spaces, as sound travels through the air and gets reflected from surfaces, high frequencies are attenuated stronger than low frequencies. To account for this effect, many artificial reverberation products feature a tone control. In the current scheme, a first order low-pass or high cut filter with adjustable corner frequency may be placed in the comb filter feedback path, refer to figure 4.

6 Discussion and Outlook

This article describes a hybrid approach to artificial reverberation, consisting of a short convolution, cross-fading, and a delay with feedback. The scheme offers a number of advantages over traditional designs:

- High echo density in both time and frequency domains, unlike some early algorithmic reverbs
- No tedious parameter tuning as in many algorithmic reverbs
- Very natural sounding reverberation tail without unpleasant resonances, as is often the case with algorithmic reverbs
- Low CPU and memory demand compared to straight convolution reverbs
- Works for all sampling rates

A ready-to-use implementation by the author is available for free [7]. Owing to its small footprint, it has found its way as an effect in various other products [8, 9].

Possible refinements such as more realistic reverberation onset, early reflections etc. may be accounted for via a direct model, while delaying the late reverberation tail accordingly.

Another optional improvement is to replace white noise by so-called velvet noise [10]. This particular noise type is perceived smoother, less grainy than straight (gaussian or other) white noise. The author's own experiments with velvet noise, however, did not suggest much difference to warrant its use in a reverb application. The argument that velvet noise results in fewer operations than straight noise may be true for direct convolution - for the block convolution used in this work it makes no difference.

References

[1] Vesa Välimäki, Julian D. Parker, Lauri Savioja, Julius O. Smith, and Jonathan S. Abel, More than 50 Years of Artificial Reverberation, AES 60th Int. Conf. Leuven, Belgium, February 3, 2016.

[2] Sean Costello, 1000 Years of Reverbs, Acoustic Engineering Society 2015, slide set.

[3] Sean Costello, 1000 Years of Reverbs, Youtube presentation, March 2019.

[4] Manfred R. Schroeder and Benjamin F. Logan, , 'Colorless' Artificial Reverberation, JAES Volume 9 Issue 3 pp. 192-197; July 1961.

[5] William G. Gardner, Efficient Convolution without Input-Output Delay, J. Audio Eng. Soc. Vol. 43, No. 3, 1995.

[6] William H. Press, Saul A. Teukolky, William T. Vetterling, Brian P. Flannery, Numerical Recipes in C, Cambridge University Press, 2nd ed., 1992, pp. 274ff.

[7] Martin Vicanek, MVerb 7B (in preparation).

[8] Rex Basterfield, Quilcom Products made with Flowsone, TOMSNARE and later.

[9] Kevin Brown, K Brown Synth Plugins, Paddy, Pad-MX, PulOddy, Vektor.

[10] Hanna Järveläinen and Matti Karjalainen, Reverberation Modeling Using Velvet Noise, 30th International AES Conference: Intelligent Audio Environments (March 2007), Paper No. 7.