EXPLORING THE VECTORED TIME VARIANT COMB FILTER

Vesa Norilo, *

Centre for Music & Technology,
Sibelius Academy, University of Arts
Helsinki, Finland
vnorilo@siba.fi

ABSTRACT

This paper presents the time variant vectored comb filter. It is an extension of the feedback delay network to time variant and nonlinear domains. Effects such as chorus and flanger, tap delay and pitch shifter are examined in the context of the feedback scheme. Efficient implementation of a stateless vectorizable LFO for modulation purposes is presented, along with a recursive formulation of the Hadamard matrix multiplication. The time variant comb filter is examined in various effect settings, and presented with source code and sound examples.

1. INTRODUCTION

A feedback delay network is a well established for method for implementing efficient synthetic reverberators. The algorithm is a simple yet elegant generalization of the comb filter; the signal and filter parameters are vectored and the feedback attenuation becomes a matrix multiplication.

Different extensions to the comb filter are also ubiquitous. The extensive design space of modulation–delay effects can be seen as comb filter variants. This leads into an intriguing possibility of further generalization, the vectored modulation delay.

This paper explores the addition of vectored delay time and amplitude modulation to the FDN. Effects resembling modulation delay staples such as chorus and flanger are examined and extended. Since all these effects are just parametrizations of the vectored time variant comb filter, various hybrids are also presented.

The fundamentals of feedback delay networks are presented first, in Section 2, *Background*. The generalization into Vectored Time Variant Comb Filters is discussed in Section 3, *Exploring the Design Space*. This section discusses the implementation and applications of the effect as well as efficient implementation of the modulation structure on vector hardware. A summary of the paper is given in Section 5, *Conclusion*.

2. BACKGROUND

The standard comb filter is shown in Figure 1. For high filter orders, it will be perceived as an echo effect. Lower order filters that result in very fast echoes are perceived as frequency response coloration. Comb filters are ubiquitous, especially in artificial reverberation. The traditional design by Schröder[1] employs a bank of these filters, tuned to generate a series of decaying echoes resembling the diffuse reverberation field.

* This work was supported by MuTri Doctoral School, Sibelius Academy, University of Arts Helsinki

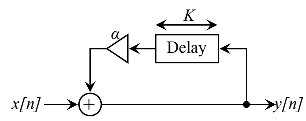

Figure 1: Comb Filter

The seminal work on feedback delay networks for artificial reverberation was done by Gerzon in the 1970s[2]. Since then, the algorithm has become a staple of synthetic reverberation. The overall schematic is similar to the comb filter: the delay and feedback coefficient are vectorized, the feedback gain stage becomes a matrix multiplication.

In contrast to the comb filter bank, each delay line feeds back into several or even all the other delay lines, giving FDN the property of an echo density that increases over time. Real acoustic spaces exhibit a similar property, unlike the constant echo density comb filter bank.

The exact nature of the FDN sound field depends on the properties of the feedback matrix. Much of the research since its discovery has been on tuning the counterintuitive algorithm. Seminal work on the subject has been done by Jot[3]. Rocchesso and Smith present important techniques and constraints for the feedback matrix design, as well as equivalences to classes of digital waveguide networks [4].

Time varying variants of the simple comb filter are also widely used. An overview of these modulation delay effects is in the literature[5]. The contribution of this paper is to explore the combination of these: the vectored, time variant comb filter, and to demonstrate an efficient implementation on SIMD hardware.

3. EXPLORING THE DESIGN SPACE

The example implementation of the vectored time variant comb filter is designed to explore the possibilities of delay and amplitude modulation of significant depth. Typically, modulation techniques in the context of feedback delay networks have been used to break the modes of the reverberator. The analogy to vectored comb filters suggests the possibility of vector chorus, vector flanger and even complicated doubler type effects. Since these are just parametrizations of the filter, hybrid effects combining features of several effects are also potentially interesting.

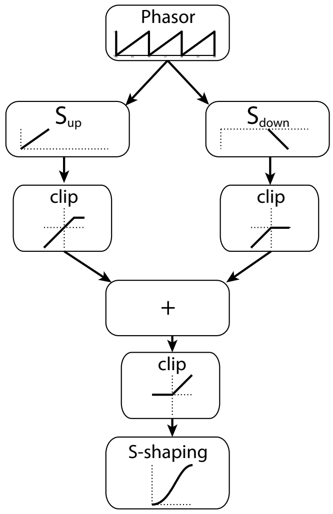

Figure 2: Overview of the waveshaping LFO

An 8-dimensional vectored time variant comb filter is implemented for the purposes of this study. The filter consists of a delay bank with loss filters and a feedback matrix. Two LFOs are provided per delay, one for delay time modulation and one for amplitude modulation.

This design should be easily adaptable for efficient processing on common SIMD units which tend to be 4 or 8 units wide at the time this article was written. The reference implementation can optionally use the Intel AVX instruction set to run most of the comb filter on a parallel SIMD code path. It should be easily adaptable to most similar vector architectures.

3.1. Implementation

3.1.1. Vectorized LFO

This section presents an LFO algorithm capable of producing control waveforms of triangle and square variety, with adjustable symmetry and slopes for ramps and pulses as well. All waveforms can be continuously morphed between linear and pseudosinusoid shape. The oscillator is designed for control signals and is not band limited.

The algorithm is designed for modern hardware and vector processing, which essentially preclude the use of nondeterministic code path or memory access. Wavetables and branch logic are thus out of question. The algorithm is a pure function waveshaper that acts on a simple phasor. Stateful or stateless phasors can be chosen according to the target hardware.

The waveshaper is presented as a cascade of stages following the phasor producing a periodic ramp in the range $[0, 1]$. An overview is given in Figure 2.

The triangle/ramp/square base shape is accomplished by two linear functions and three clipping stages. The base waveform is parametrized by three degrees of freedom, (x_1, x_2, x_3), as shown in Figure 3. The linear functions for S_{up} and S_{down} follow trivially from these points and are given in Equations 1 and 2. For the linear segments to be defined, $x_1 > 0 \wedge x_2 > x_1 \wedge x_3 > x_2$. How small the deltas can be depends on the numerical characteristics of the target hardware.

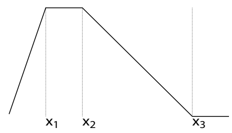

Figure 3: LFO degrees of freedom

The waveform is combined by clipping S_{up} below one and S_{down} below zero. Summing these and clipping above zero yields the final waveform in the unipolar range of $[0, 1]$. S_{down} should be computed in the form $k(x - x_2)$ to preserve numerical precision near zero – the section that will actually be used.

A pseudo-sinusoid waveform can be accomplished by a further waveshaping polynomial (Equation 3). This shaping turns the linear segments in the LFO into S-shape curves that are continuous in the first derivative when applied to a triangle-like wave. A continuous control parameter from linear to pseudo-sinusoid segment can be introduced. All in all, the pseudo-sinusoid shape morphing roughly doubles the computational complexity of the LFO. The S-curve is potentially useful for all of the waveforms: triangle, skewed pulse and ramp.

$$S_{up}(x) = \frac{x}{x_2 - x_1} \tag{1}$$

$$S_{down}(x) = \frac{x - x_2}{x_2 - x_3} \tag{2}$$

$$h(x) = 6(\frac{x^2}{2} - \frac{x^3}{3}) \tag{3}$$

3.1.2. Delay and Filter Bank

The delay and attenuation filter banks used in the effect are straightforward. The filter bank is based on the standard first order loss filter. The delays are implemented as circular buffers.

The one pole filter bank is an easy fit for vector hardware. The same can not be said for the delay bank, due to non-uniform ring buffers. This leads to the memory access pattern requiring a scatter/gather idiom.

The modulation of delay lines makes the signal path nonlinear. This undermines the canonical stability criteria for feedback delay networks. The described modulation is attenuation rather than boost, so an unstable situation is not expected. However, the attenuation effect of amplitude modulation is unpredictable and program dependent. That is why a manual adjustment of feedback beyond 100%, is provided per delay line, with total stability guaranteed by an additional stage for soft saturation.

3.1.3. Feedback Matrix

As in reverberators, a lossless feedback matrix is the starting point. Such matrices are unitary. For the purpose of this study, the orthogonal Hadamard matrix with computationally beneficial features is used. The matrix is generated by taking a N-fold Kronecker product of seed matrices and scaling for orthogonality, as shown in Equation 4. This matrix caters for a network of 2^N delay lines.

$$H_n = \frac{1}{\sqrt{2^N}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}^{\otimes N} \tag{4}$$

Table 1: Permute–flip–add sequence for 8×8 Hadamard matrix

permute a	0	0	2	2	4	4	6	6
permute b	1	1	3	3	5	5	7	7
sign	+	-	+	-	+	-	+	-
permute a	0	1	0	1	4	5	4	5
permute b	2	3	2	3	6	7	6	7
sign	+	+	-	-	+	+	-	-
permute a	0	1	2	3	0	1	2	3
permute b	4	5	6	7	4	5	6	7
sign	+	+	+	+	-	-	-	-

In the case of two delay lines ($N = 1$), the matrix computation trivially results in a vector containing their sum and difference. A larger feedback matrix can be constructed by computing pairwise feedback vectors, then recursively combining pairs of them by concatenating the vector sum and difference. Each level of recursion corresponds to a Kronecker product, doubling the number of diffuse feedback channels. This algorithm results in $N2^N$ additions, or by the number of delay lines, $nlog_2(n)$ additions as noted in the literature for FDN reverberators[6]. The matrix scaling coefficient $\frac{1}{\sqrt{2^N}}$ can be integrated into the delay line loss filters.

The feedback matrix is also amenable to SIMD computation. Each Kronecker product can be reduced to a vectored permute, sign flip and addition. An example with $N = 3$ is demonstrated in Table 1. Three products are shown. The permute rows a and b correspond to element indices for the left and right hand side of the addition; the sign row denotes sign flips for the right hand side. For architectures with a vector width of 2^N, the entire feedback matrix can be computed in $4N$ vector operations, corresponding to two permutes, sign flip (*xor*) and addition per Kronecker product.

Alternatively, the Hadamard matrix could be vectorized as a time-parallel computation in block processing. This choice could be considered as it saves the permute operations described above; however, it is less appealing due to the matrix appearing in a feedback loop of a modulation delay. The various techniques to work around the latency of such an algorithm would likely cost more than the simple permutation instructions, both in terms of compute efficiency and algorithmic complexity.

3.1.4. Control Surface and Parameter Mapping

The internal parameter set used for each delay line in the effect is shown in Table 2. A one to one mapping from the internal parameter set to a user interface is not likely very attractive. For eight delay lines, the interface would contain 144 parameters. Macro controls would be more useful; this should be studied in the future.

3.2. Applications and Qualitative Evaluation

This section briefly discusses some of the creative possibilities of using the effect described in this study. The evaluations are the subjective impressions of the author; they shouldn't be read as scientific results. For a more detailed perspective, please refer to the example code and sound files that are available at the code repository specified at the conclusion of this paper.

Table 2: Time Variant Vectored Comb parameter set

parameter	**unit**	**description**
delay	ms	delay time
in gain	dB	input signal to delay line
out gain	dB	delay line to output signal
out pan	linear	stereo panorama
tone	linear	loss filter to feedback matrix
fb gain	%	delay line to feedback matrix
LFO rate	Hz	
DM depth	ms	delay time modulation
DM offset	linear	DM phase offset
DM shape	$4\times$ linear	$x_1, x_2, x_3, shape$
AM depth	linear	amplitude modulation
AM offset	linear	AM phase offset
AM shape	$4\times$ linear	$x_1, x_2, x_3, shape$

3.2.1. Vector Chorus–Flanger

The vectored chorus–flanger revolves around delay times and delay modulation depth of $0 - 40$ ms and LFO rates in the range of $0.1 - 5$ Hz. Adding feedback creates a flanger-like moving resonance effect, but the tonal color is a lot more complicated as the feedback network system has a large number of poles.

With longer delay times and feedback, the effect acquires spring reverb characteristics, especially with faster LFO rates.

Complex stereo imaging can be achieved by using variations of similar settings on multiple delay lines and panning them across the image.

3.2.2. Multitap Delay

By using delay times a lot longer than those in a diffuse field reverberator, a sparse multitap delay effect is created. What is especially interesting is the echo density escalation over time. The sparse echoes gradually become a diffuse tail. The effect is capable of generating interesting transitions from percussive textures to static ones.

3.2.3. Pitch Shifter

By using ramp-shape delay time modulation together with phase shifted triangular amplitude modulation results in a simple pitch shifter. The ramp modulator adjusts momentary playback speed, while the triangular envelope is aligned to hide the discontiuity in the ramp. An even overrall amplitude can be attained by using overlapping shifters with orthogonal phase shifts.

The complex feedback is the distinguishing feature from standard slicer shifters. The effect is less useful as a plain transposition, as an infinite number of high order transpositions are generated by the feedback, but can result in extremely full ensemble thickening effects with subtle pitch shift factors.

3.2.4. Hybrid Effects

Interesting combinations of effects can be realized by mixing delay line settings from several of the above categories. Reverb-like settings together with pitch shifter and chorus effects appear to be the most immediately useful.

3.2.5. Semi-Stable Self Oscillation

By utilizing very short delay times in the range of $0 - 20$ ms and feedbacks in excess of 100%, a self-oscillating network can be created. Soft saturation in the feedback loop prevents the blow-up and introduces both harmonics and non-harmonic aliasing frequencies. The tonalities due to complex feedback paths are interesting, but the pitch is quite hard to predict and control.

4. PERFORMANCE EVALUATION

4.1. LFO

The performance of the LFO reference implementation was measured by accumulating its output over 100 000 000 sample frames to ensure timing accuracy. The accumulator is in place to prevent dead code optimization by the compiler. A vectorized LFO with eight independent waveforms was measured. The test program was compiled with Microsoft Visual Studio 2013, with AVX architecture and the fast floating point model enabled. The measurement was run on Windows 7 with the dual core Intel i5-3317U CPU clocked at 1.70GHz.

The vectorized LFO was able to produce 1.14286×10^9 output frames of 8 discrete signals per second – roughly 16 CPU cycles per frame, or 2 cycles per sample. This translates to a real time CPU core utilization of 0.0039% per modulation signal on the machine the measurement was performed on, when processed at 44.1kHz.

4.2. Time Variant Vectored Comb Filter

The entire effect consists of the following modules:

1. delay bank (8)
2. modulation LFOs (16)
3. loss filter bank (8)
4. feedback matrix (8×8)
5. input and output routing matrices

The loss filter, modulation LFOs and all gain and summation stages trivially vectorize to SIMD code. The feedback matrix is also fully vectorized, as explained in Section 3.1.3. The modulation delay bank remains scalar, as the requisite scatter/gather operations defeat the purpose of vectorization on current hardware.

In a test harness like the one described in 4.1, the entire comb filter implementation produced 3.57×10^6 stereophonic output frames per second. This translates to a real time CPU core utilization of 1.23% at 44.1kHz. Roughly 80% of the time is spent in the scalar delay line bank. This suggests that the additional computational load from a comprehensive feedback matrix, in contrast to a plain modulation delay bank, is far from prohibitive in the context of suitable vector hardware.

5. CONCLUSION

This paper examined the extension of feedback delay networks into the realm of modulation delay effects. Efficient vectorized implementation of the parallel modulation structure and a diffusive feedback matrix were demonstrated.

The generalized time variant vectored comb filter is interesting in the sense that it is a superset of a large number of delay-based effects. It offers musically relevant and divergent possibilities when complicated feedback structures are used. In particular, hybrids between spatial and ensemble effects offer novel sounds. Continuous morphing from one effect state to another is also easily attainable.

Possible future work could involve a deeper investigation of the feedback matrix. The current implementation uses a fixed feedback matrix for maximum efficiency. Control over the diffusion between the submatrices of the Hadamard tree could be especially interesting, as it could be seen as a way to isolate or combine sections of the network. The impact of advancing scatter/gather implementations could be interesting in improving the performance of the scalar delay bank. The user interface is also an open question: the exposal of the extensive parameter set via a higher level control surface could increase the viability of the effect from the end user point of view.

A reference implementation of the effect in C++, along with sound examples, is available on Bitbucket under the MIT license, at `https://bitbucket.org/vnorilo`.

6. ACKNOWLEDGEMENTS

The work of the author was supported by the MuTri doctoral school, Sibelius Academy, University of Arts Helsinki. The vector image in Figure 1 is sourced from WikiMedia under the creative commons public domain license.

7. REFERENCES

[1] M R Schroeder, “Digital Simulation of Sound Transmission in Reverberant Spaces,” *Journal of the Acoustical Society of America*, vol. 45, no. 1, pp. 303, 1969.

[2] M A Gerzon, “Unitary (energy-preserving) multichannel networks with feedback,” *Electronics Letters*, vol. 12, no. 11, pp. 278–279, 1976.

[3] Jean-Marc Jot and Antoine Chaigne, “Digital Delay Networks for Designing Artificial Reverberators,” in *the 90th AES Convention*, 1991, vol. 3030, p. preprint no. 3030.

[4] Davide Rocchesso and Julius O Smith, “Circulant and elliptic feedback delay networks for artificial reverberation,” *Speech and Audio Processing, IEEE Transactions on*, vol. 5, no. 1, pp. 51–63, 1997.

[5] P Dutilleux, M Holters, Sascha Disch, and Udo Zölzer, “Filters and delays,” in *DAFX: Digital Audio Effects*, Udo Zölzer, Ed., pp. 47–82. John Wiley & Sons,Inc., 2nd edition, 2011.

[6] Davide Rocchesso, “Maximally Diffusive Yet Efficient Feedback Delay Networks for Artificial Reverberation,” *IEEE Signal Processing Letters*, vol. 4, no. 9, pp. 252–255, 1997.