

## EXTENSIONS AND APPLICATIONS OF MODAL DISPERSIVE FILTERS

Elliot K. Canfield-Dafilou and Jonathan S. Abel

Center for Computer Research in Music and Acoustics,  
Stanford University, Stanford, CA 94305 USA  
kermit|abel@ccrma.stanford.edu

### ABSTRACT

Dispersive delay and comb filters, implemented as a parallel sum of high-Q mode filters tuned to provide a desired frequency-dependent delay characteristic, have advantages over dispersive filters that are implemented using cascade or frequency-domain architectures. Here we present techniques for designing the modal filter parameters for music and audio applications. Through examples, we show that this parallel structure is conducive to interactive and time-varying modifications, and we introduce extensions to the basic model.

### 1. INTRODUCTION

Dispersion filters, in which the various frequency components of an input signal are delayed by different amounts, find widespread use in audio processing. Often they are used to emulate the dispersive characteristic of physical systems. For example, [1] proposes a method for designing high-order allpass filters which were applied in [2] to model spring reverberators where low frequencies propagate faster than high frequencies and in [3] to model the dispersion of stiff strings where the high frequencies travel faster.

Other times, dispersive filters are designed to compensate for unwanted frequency-dependent delay. For example, [4–6] propose methods for designing allpass filters to equalize the group delay of elliptic filters. Recently, [7–9] and others have shown applications of dispersive filters for delay compensation between multiple drivers in a loudspeaker.

Rather than compensating for unwanted delay, some situations call for dispersive systems that add frequency-dependent delay with specific characteristics. For example, [10, 11] propose methods for decorrelating audio signals using high-order allpass systems. Additionally, [12, 13] have proposed using high-order dispersive systems for abstract sound synthesis and processing.

In many applications, the dispersion filters are of very high order, having dozens to hundreds of poles. They are often implemented as high-order difference equations or biquad cascades. In the case of dispersion filters having thousands of poles, DFT-based convolution with the associated impulse response has been used [10]. A drawback to both high-order difference equations and biquad cascades is that they are prone to numerical difficulties. While biquad cascades are robust compared with high-order difference equation implementations, numerical errors accumulate through the cascade. DFT-based convolution techniques can produce high-order systems with large amount of dispersive delay, however they add latency.

Copyright: © 2019 Elliot K. Canfield-Dafilou and Jonathan S. Abel. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Another drawback to both high-order difference equations and biquad cascades is that it is difficult to interactively change the desired group delay,  $\tau(\omega)$ . First, designing new filter coefficients to produce the new desired dispersive delay may be computationally costly. Second, substituting the new coefficients in the IIR filters is difficult, as the substitution interacts with the filter state, likely producing unwanted artifacts while the change ripples through the system. Third, certain changes, such as increasing or decreasing the number of coils in a simulated spring reverberator or modifying the length of a modeled string, change the number of poles needed to implement the desired dispersion—something that is difficult to do without artifacts. Frequency-domain implementations also present real-time interaction difficulties. Since frequency-domain methods process data in blocks, they produce computational latency precluding sample-by-sample processing and real-time interaction.

We recently proposed a modal approach for designing and implementing dispersive systems [14] that uses the modal architecture described in [15]. Two modal dispersion filters were introduced: a modal comb filter with multiple dispersive arrivals, and a modal delay filter with a single dispersive arrival. As will be shown in this work, the parallel structure is conducive to interactive modification of the dispersive characteristics and avoiding numerical issues associated with other methods. This paper will focus on time-varying dispersive audio effects and other extensions for modal dispersive filters.

### 2. DISPERSION FILTER DESIGN

We use a modal architecture, as shown in Fig. 1b to implement dispersive delay and comb filters. Its system impulse response, denoted by  $h(t)$ , is the sum of  $M$  parallel resonant filters with mode responses  $h_m(t)$ ,  $m = 1, 2, \dots, M$ ,

$$h(t) = \sum_{m=1}^M h_m(t). \quad (1)$$

The resonant mode responses  $h_m(t)$  are complex exponentials, each characterized by a mode frequency  $\omega_m$ , mode damping  $\alpha_m$ , and complex mode amplitude  $\gamma_m$ ,

$$h_m(t) = \gamma_m e^{(j\omega_m - \alpha_m)t}. \quad (2)$$

The system output  $y(t)$  in response to an input  $x(t)$  is then seen to be the sum of mode outputs

$$y(t) = \sum_{m=1}^M y_m(t), \quad y_m(t) = h_m(t) * x(t), \quad (3)$$

where the  $m$ th mode output  $y_m(t)$  is the  $m$ th mode response convolved with the input.

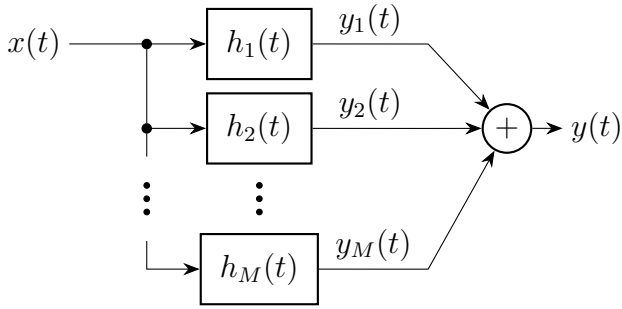


Figure 1: Modal filter architecture consisting of a parallel combination of resonant mode filters.

In the remainder of this section, we will show how to set the modal parameters to implement a desired dispersive filter. The desired delay  $\tau(\omega)$  is used to specify the number of modes  $M$  and the mode frequencies  $\omega_m$ . The desired decay time  $T_{60}(\omega)$  and desired magnitude equalization  $q(\omega)$  are used to fix the mode dampings  $\alpha_m$  and mode amplitudes  $\gamma_m$ .

We begin by developing the formulation for dispersive comb and delay filters through Fourier theory.

## 2.1. Derivahrough Fourier Transform

Consider the  $M$ -point Inverse Discrete Fourier Transform

$$x(n) = \frac{1}{M} \sum_{m=-M/2}^{M/2-1} X(m) e^{j\frac{2\pi mn}{M}}, \quad (4)$$

where  $X(m)$  represents the coefficients of  $M$  basis frequencies, indexed by  $m$ , and where  $n$  is the discrete time index. In the case where the coefficients  $X(m)$  are independent of frequency (e.g.,  $X(m) = M$ ),

$$x(t) = \sum_{m=-M/2}^{M/2-1} e^{j\frac{2\pi m f_s}{M} \frac{n}{f_s}} = \sum_{m=-M/2}^{M/2-1} e^{j\omega_m t}. \quad (5)$$

The time domain signal is a band-limited, sampled, periodic sinc function with peaks every  $M$  samples as seen in Fig. 2a. We have introduced a sampling rate  $f_s$  which allows us to write angular frequency in radians per second,  $\omega_m = j2\pi m f_s / M$ , and time in seconds,  $t = n / f_s$ .

If we double  $M$  while maintaining the same sampling rate, meaning we double the frequency density of sinusoidal basis functions, the sinc has twice the period in the time domain as seen in Fig. 2b. Following this logic, the delay at each frequency is proportional to the frequency density of sinusoidal basis functions. Instead of a pure delay, a dispersive system can be formed by setting the frequencies of the sinusoidal bases according to the desired frequency-dependent delay  $\tau(\omega)$ .

To further unite this derivation of dispersive delay filters with our modal implementation, we introduce two more concepts. First, we introduce a damping factor  $\alpha$  that causes the signal to subside over time,

$$x(t) = \left( \sum_{m=-M/2}^{M/2-1} e^{j\omega_m t} \right) e^{-\alpha t}, \quad (6)$$

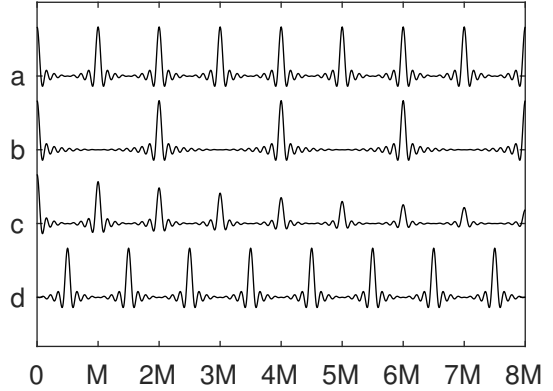


Figure 2: An upsampled sinc function showing the periodicity of the DFT (a). When the length of the DFT is doubled, the delay doubles (b). (c) shows the inclusion of the damping filter and (d) shows the effect of the shift theorem.

as seen in Fig. 2c. This damping can also be frequency-dependent  $\alpha_m$  and factored inside the sum,

$$x(t) = \sum_{m=-M/2}^{M/2-1} e^{(j\omega_m - \alpha_m)t}. \quad (7)$$

Additionally, we can use the shift theorem,

$$x(t - \Delta) \longleftrightarrow e^{-j\omega_m \Delta} X(m), \quad (8)$$

to modify the time of the initial delay arrival. For example, we can achieve an odd integer set of arrivals by introducing a phase term of  $e^{j\pi m} = (-1)^m$  into (5), as seen in Fig. 2d.

## 2.2. Dispersive Comb Filter Design

Following [14], the number of modes  $M$  is the number of samples of delay, averaged across the band from DC to the Nyquist limit,  $f_s/2$ ,

$$M = \sum_{n=0}^N \frac{\tau_n f_s}{2N}, \quad (9)$$

where  $\tau_n$ ,  $n = 0, 1, \dots, N$ , represents the desired delay  $\tau(\omega)$  evaluated at the  $N$  discrete frequencies  $\tau(2\pi n f_s / 2)$ , and where  $M$  can be rounded or otherwise adjusted to be an integer.

The mode frequencies  $\omega_m$  are chosen to be those frequencies at which the cumulative delay  $\varphi(\omega_m)$  hits integer multiples of  $2\pi$ ,

$$\varphi(\omega_m) = \int_0^{\omega_m} \tau(\nu) d\nu = 2\pi m. \quad (10)$$

The mode dampings  $\alpha_m$  may be set according to a desired 60 dB decay time as a function of frequency  $T_{60}(\omega)$ ,

$$\alpha_m = \frac{\ln(0.001)}{T_{60}(\omega_m)}. \quad (11)$$

Alternatively,  $\alpha_m$  may be set to have a 60 dB decay after a given number of arrivals,  $N_{60}(\omega)$ . We then have

$$\alpha_m = \frac{\ln(0.001)}{(2N_{60}(\omega_m) - 1) \cdot \tau(\omega_m)}, \quad (12)$$

where the factor  $2N_{60}(\omega_m) - 1$  is used assuming arrivals at odd integer multiples of the designed arrival time  $\tau(\omega_m)$ .

Since the energy in a given mode is proportional to its decay time, we set the mode gains according to

$$\gamma_m = \frac{e^{j\theta m} \alpha_m}{\tau(\omega_m)} \cdot q(\omega_m), \quad \theta \in [0, 2\pi), \quad (13)$$

where  $\theta$  controls the initial time delay according to the shift theorem (8), the denominator  $\tau(\omega)$  performs an allpass equalization, and  $q(\omega)$  provides the desired frequency equalization. Unless otherwise stated, we use  $\theta = \pi$  for odd integer multiples of the designed delay in this paper.

An example dispersive comb filter designed to have regions consisting of both smooth and staircase dispersion is shown in Fig. 3. The first arrival of the desired frequency-dependent spectral delay is shown as a dotted line overlaid on the spectrogram. Note that the impulse response’s first arrival closely tracks the target time delay, and the subsequent arrivals are seen to have the anticipated odd integer multiples of the designed dispersive time delay. Fig. 4 shows the same dispersive characteristic as Fig. 3, but designed to decay 60 dB after eight arrivals.

### 2.3. Dispersive Delay Filter Design

There are two approaches for converting a dispersive comb filter into a dispersive delay filter that has only one arrival. First, the mode decay rates could be set to achieve a significant amount of attenuation between arrivals. For a system with odd integer multiples of the desired delay, we design the dampings to produce  $\lambda$  dB of decay between successive arrivals,

$$\alpha_m = \frac{\ln(10^{-\lambda/20})}{2\tau(\omega_m)}, \quad (14)$$

and scale the mode gains by  $\lambda/2$  dB. This means the first dispersive arrival will have a roughly unit level and subsequent arrivals will be attenuated by at least  $\lambda$  dB. For audio applications,  $\lambda$  in the range 60–80 dB would render the unwanted subsequent arrivals inaudible. For instance, an attenuation of  $\lambda = 60$  dB was used to design the dispersive delay shown in Fig. 5.

Since the  $\lambda/2$  dB gain could create numerical difficulties, an alternative approach is to use a truncated IIR (TIIR) filter [16] to eliminate the unwanted subsequent echos as described in [14].

## 3. EXTENSIONS

In addition to the dispersive delay and comb filters shown above, the parameters exposed by the modal framework provide a powerful resource additional modification. Throughout this section, we will show the powerful effects of some simple modifications to the modal parameters and show some time-varying examples.<sup>1</sup>

When implementing time-varying filters it is important that the filters remain stable and avoid irritating artifacts that may arise from changing parameters quickly. We use Max Mathews’s phasor filter [17] to implement these modal dispersion filters as seen in (3). This filter uses the property that when complex numbers are multiplied together, the magnitude is the product of their magnitudes and the phases sum [18]. With this implementation and

<sup>1</sup>Audio examples associated with the figures in the paper can be found at <https://ccrma.stanford.edu/~kermit/website/ddf.html>

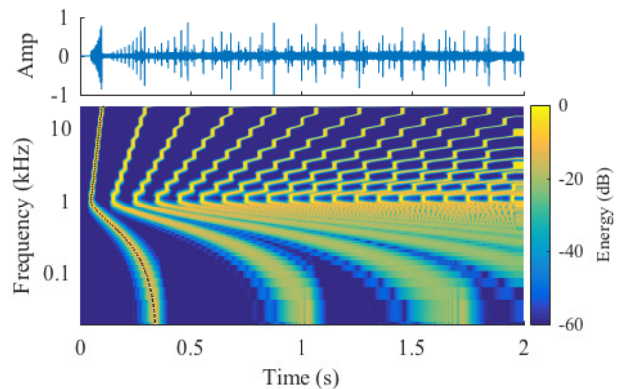


Figure 3: Impulse response (top) and spectrogram (bottom) of an example dispersive comb filter. The desired dispersive delay  $\tau(\omega)$  is shown as a dotted line overlaid on the spectrogram.

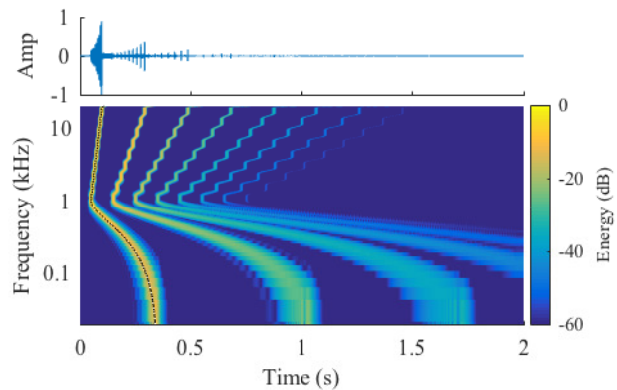


Figure 4: Impulse response (top) and spectrogram (bottom) of the dispersive comb filter from Fig. 3, set to decay 60 dB after eight arrivals.

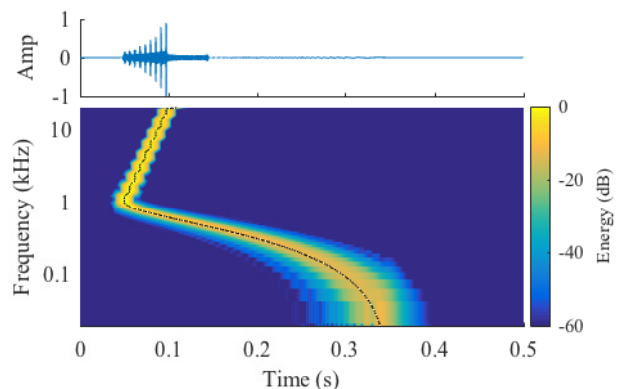


Figure 5: Impulse response (top) and spectrogram (bottom) of a dispersive delay filter constructed using decay time approach and the same designed delay as Figs. 3 and 4.

the parallel structure, the dampings, frequencies, amplitudes, and input level can all be functions of time without introducing transients or other artifacts when the filter state is changed since the state of each filter does not depend on the state of the others.

### 3.1. Modifying the Number of Necessary Modes

The number of modes necessary to implement a dispersive filter is equivalent to the average delay as described by (9). In certain situations, it could be desirable to implement one of these filters with a specific number of modes (e.g., running with limited computational resources). Modifying the number of modes, however, comes at the expense of distorting the amount of delay at each frequency. A target group delay implemented with a desired number of modes  $\tau_M(\omega)$  can be found by normalizing the old group delay by the average delay and scaling the result by the desired number of modes  $M$ ,

$$\tau_M(\omega) = M \frac{\tau(\omega) + k}{\sum_{n=0}^N \frac{(\tau_n + k)f_s}{2N}}, \quad k > -\min_t \tau(\omega), \quad (15)$$

where  $\tau_n$ ,  $n = 0, 1, \dots, N$ , represents the desired delay  $\tau(\omega)$  evaluated at the  $N$  discrete frequencies  $\tau(2\pi n f_s/2)$  and  $k$  represents added delay that is independent of frequency. This additional delay controls how the  $M$  modes are distributed in frequency. When  $k = 0$ , the entire group delay curve is scaled by the amount necessary to have  $M$  samples of average delay. As  $k$  approaches  $-\min_t \tau(\omega)$ , any constant delay in  $\tau(\omega)$  will be eliminated and the frequency regions with the most delay will be exaggerated. As  $k$  becomes large, the detail of the group delay will be reduced until it is constant across frequency. Fig. 6 shows an example group delay curve warped with different values of  $k$ .

### 3.2. Transitioning Between Delay Characteristics

The amount of delay in a local frequency neighborhood is proportional to the density of modes in that neighborhood.

If we want to transition between two delay characteristics that have the same average delay,  $\tau_a(\omega)$  and  $\tau_b(\omega)$ , we can interpolate between their mode frequencies over time. Because the mode frequencies will move, causing the delay to change, we will observe some Doppler shift during the transition. This may or may not be desirable. An example can be seen in Fig. 7

Another scheme for transitioning between delay characteristics can be accomplished by computing the output of the mode filters of both delay trajectories simultaneously, and crossfading the mode amplitudes. Here there will be no pitch shift, however during the transition, both dispersive characteristics will be audible simultaneously. Fig. 8 shows an example amplitude crossfade using the same dispersion filters and transition time as compared to Fig. 7. Fig. 9 shows a guitar track processed by dispersive comb filters with time-varying mode frequencies.

If the average delay is different (i.e., the number of modes is not the same), or we want to prevent mode frequencies from moving beyond a prescribed amount, we need a scheme for “birth and death” of mode filters [19]. This can be accomplished by using the mode amplitudes to fade “new” modes in and fade out “dead” modes in combination with amplitude and/or frequency morphing.

### 3.3. Damping Modifications

It is trivial to lengthen or shorten the decay time associated with each mode. The number of echos in each frequency band can be

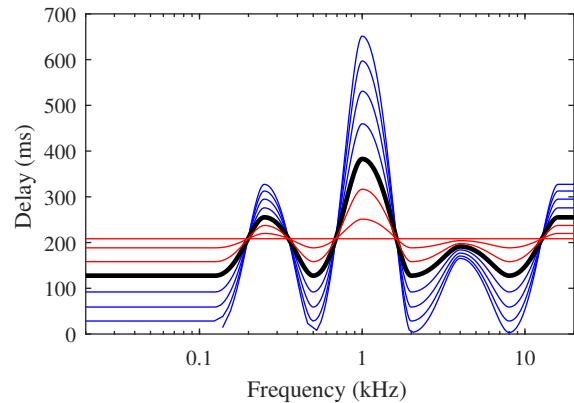


Figure 6: A set of group delays  $\tau(\omega)$  that have the same average delay (i.e. the same number of modes) using (15) and different values of  $k$ —black:  $k = 0$ ; red:  $k > 0$ ; blue:  $k < 0$ .

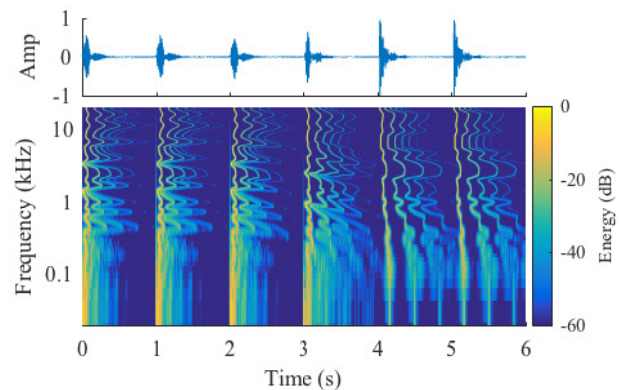


Figure 7: Time domain (top) and spectrogram (bottom) of an impulse train processed by a dispersive comb filter with time-varying mode frequencies. Note that the dispersive characteristic changes with the local mode density and some pitch shifting occurs.

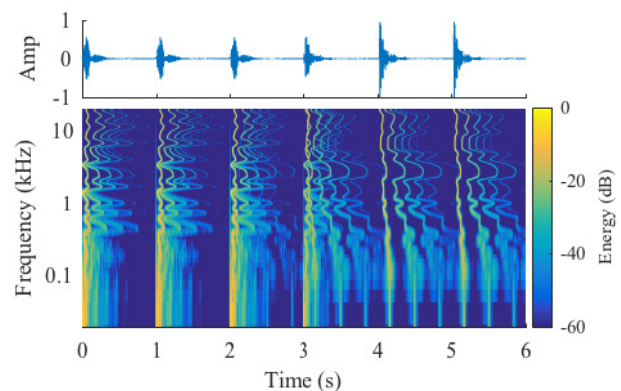


Figure 8: Time domain (top) and spectrogram (bottom) of an impulse train processed by two dispersive comb filters using amplitude modifications to cross-fade between the dispersive characteristics. Note that during the transition, both dispersive characteristics are audible and visible in the spectrogram.

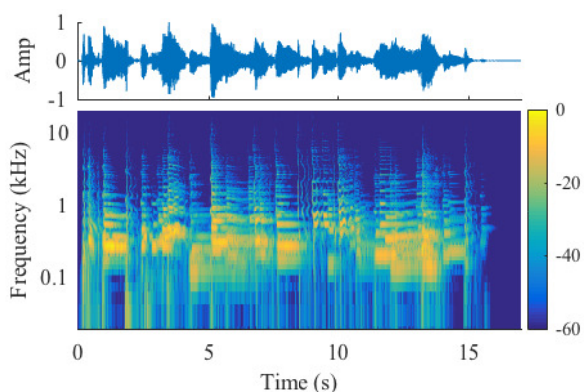


Figure 9: Time domain (top) and spectrogram (bottom) of a guitar track processed with dispersive comb filters with time-varying mode frequencies.

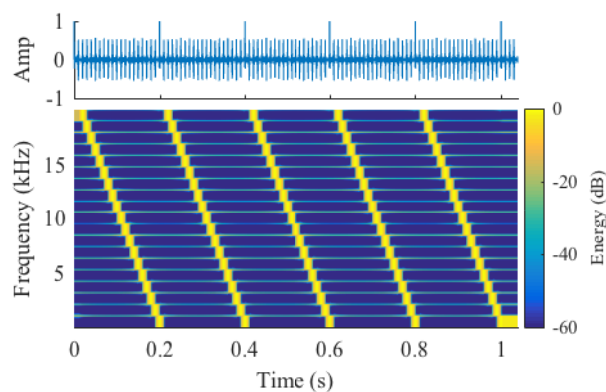


Figure 12: Time domain (top) and spectrogram (bottom) of a dispersive comb filter set to have a constant repeat rate and piecewise linear phase shift as a function of frequency.

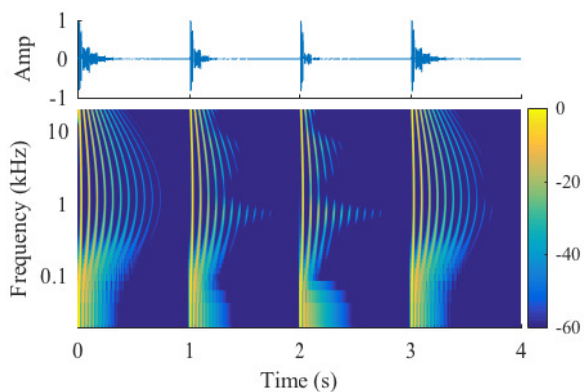


Figure 10: Time domain (top) and spectrogram (bottom) of an impulse train processed by a dispersive comb filter set to have time-varying scaling applied to the damping coefficients.

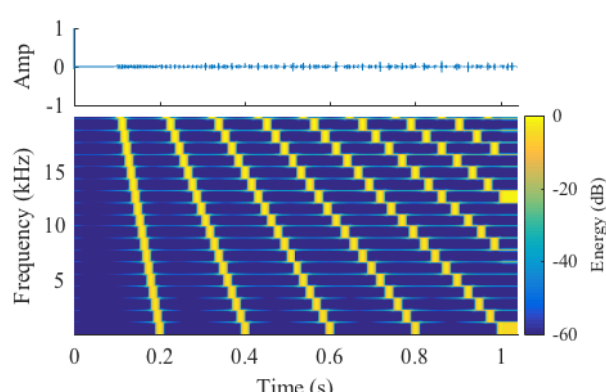


Figure 13: Time domain (top) and spectrogram (bottom) of a dispersive comb filter set to have a piecewise frequency-dependent delay to match the second reflection of Fig. 12. Notice that the lowest frequencies repeat on the same time interval in both figures while short initial delay time of the high frequencies causes a shorter period between subsequent arrivals.

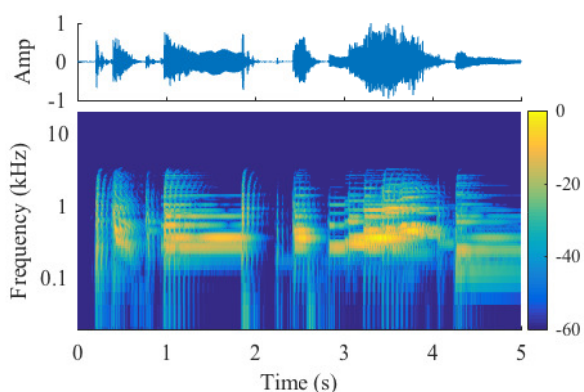


Figure 11: Time domain (top) and spectrogram (bottom) of a guitar track processed by a dispersive comb filter set to have damping coefficients that vary according to the input level of the guitar signal. The dispersion filter models a spring reverb where the reverb lasts longer when the input signal is louder.

frequency-dependent or be allowed to vary over time. For example, a dispersive comb filter could be designed that decays more quickly in the high frequencies than the low frequencies. Fig. 10 shows an example dispersive filter where the damping factors are frequency-dependent and time-varying.

Alternatively, a level tracker could be employed to modulate the number of echos that appear in the output based on the amplitude of the input signal. Fig. 11 shows a guitar track processed by a dispersive filter that reacts to its tracked level.

### 3.4. Phase Modifications

At each mode frequency a pulse is observed at repeated intervals that depend on the the desired delay  $\tau(\omega)$  and the phase angle  $\theta$ . We can have multiple frequency-dependent initial time offsets by allowing the phase angle to also be a function of frequency  $\theta(\omega)$ . As an example, Fig. 12 shows a filter constructed to have piecewise

constant phase on the interval  $[0, 2\pi)$  such that all frequencies have the same repeat rate in time but a different initial phase. As a comparison, Fig. 13 models the same second dispersive arrival as frequency-dependent delay rather than with phase modifications. As a result, the periodicity of the subsequent arrivals is frequency-dependent.

If the phase of each mode is randomized, the result is a noisy signal constrained in time by the decay rates. This signal is akin to the description of late-field reverberation synthesis described in [15]. By allowing the phase to vary with time, and smoothly change between a coherent phase (where  $\theta$  is constant) and a random phase (where each mode output is rotated by an independent, unit magnitude complex number), we can morph between dispersive delay and reverberant effects, such as seen in Fig. 14.

### 3.5. Echos Subsiding to Noise

Instead of a dispersive comb with clear, decaying echos, it is sometimes desirable to have a comb filter where each subsequent echo is a little more diffuse. After some number of echos, the signal is a noise-like wash where individual reflections are no longer detectable. The idea here is to perturb the frequencies of the modes by an amount small enough to be initially inaudible but cause the succeeding echos to be more spread out in time. The bandwidth of the perturbation is proportional to the number of desired audible echos. To have  $p$  distinct arrivals audible above the noise-like wash, the mode frequencies  $\omega_m$  should be perturbed by noise with standard deviation of  $1/p$  of the local frequency difference,

$$\tilde{\omega}_m = \frac{\omega_{m+1} - \omega_{m-1}}{2p} \nu_m + \omega_m, \quad (16)$$

where  $\nu_m$  is a sample of zero-mean unit-variance noise, e.g. having a Gaussian or triangular distribution. The small perturbations are amplified with each subsequent arrival creating the desired effect. Fig. 15 shows an example dispersion filter, compared to Fig. 16 where the frequencies were perturbed to cause the echos to turn into noise after five reflections. This processing has applications for reverberation type effects.

## 4. CONCLUSION

Dispersive filters have music and audio applications ranging from physical modeling of dispersive systems to abstract sound synthesis. In this work, we explored extensions and applications of the modal dispersive delay and comb filters introduced in [14].

We began by showing how the modal formulation of these dispersion filters can be interpreted through Fourier theory. We then described how to set the modal parameters to achieve a desired dispersion characteristic based on the modal frequency density. Following that, we showed how the parallel structure of the modal architecture and the numerical properties of the phasor filter make it possible to efficiently and interactively modify the properties of these dispersion filters. Unlike frequency-domain or cascade architectures, it is simple to implement time-varying dispersion effects using the modal approach.

We showed a range of simple modifications for the modal parameters, and backed with examples, demonstrated an assortment of musical uses of dispersive comb and delay filters. Even so, there are certainly many more ways to extend this flexible structure, such as incorporating the pitch, time, and distortion processing described in [20] with the approaches presented here.

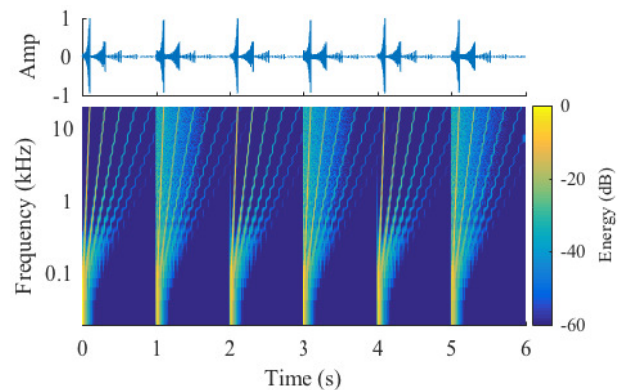


Figure 14: Time domain (top) and spectrogram (bottom) of an impulse train processed by a dispersive comb filter set to have periodic phase synchronization/desynchronization.

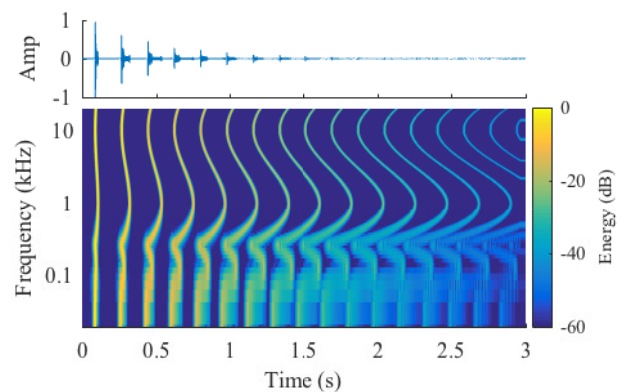


Figure 15: Time domain (top) and spectrogram (bottom) a dispersive comb filter without frequency perturbation.

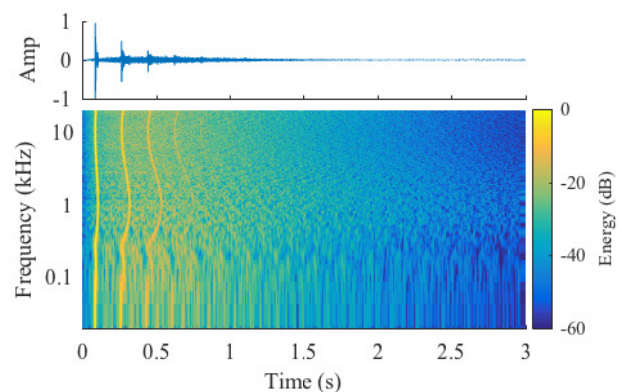


Figure 16: Time domain (top) and spectrogram (bottom) a dispersive comb filter with frequency perturbation designed to transition to noise after 4 arrivals.

## 5. REFERENCES

- [1] Jonathan S. Abel and Julius O. Smith III, “Robust design of very high-order allpass dispersion filters,” in *Proceedings of the 9th International Conference on Digital Audio Effects*, 2006, pp. 13–8.
- [2] Jonathan S. Abel, David P. Berners, Sean Costello, and Julius O. Smith III, “Spring reverb emulation using dispersive allpass filters in a waveguide structure,” in *Proceedings of the 121st Audio Engineering Society Convention*, 2006.
- [3] Jonathan S. Abel, Vesa Välimäki, and Julius O. Smith III, “Robust, efficient design of allpass filters for dispersive string sound synthesis,” *IEEE Signal Processing Letters*, vol. 17, no. 4, pp. 406–9, 2010.
- [4] Zhongqi Jing, “A new method for digital all-pass filter design,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 35, no. 11, pp. 1557–64, 1987.
- [5] Piotr Okoniewski and Jacek Piskorowski, “An analytical approach to the group delay compensation of digital IIR filters,” in *Proceedings of the 17th IEEE International Conference on Methods and Models in Automation and Robotics*, 2012, pp. 75–8.
- [6] Mauricio F. Quélhas, Antonio Petraglia, and Mariane R. Petraglia, “Efficient group delay equalization of discrete-time IIR filters,” in *Proceedings of the 12th IEEE European Signal Processing Conference*, 2004, pp. 125–8.
- [7] Aki Mäkivirta, Juho Liski, and Vesa Välimäki, “Effect of delay equalization on loudspeaker responses,” in *Proceedings of the 144th Audio Engineering Society Convention*, 2018.
- [8] Shintaro Hosoi, Hiroyuki Hamada, and Nobuo Kameyama, “An improvement in sound quality of LFE by flattening group delay,” in *Proceedings of the 116th Audio Engineering Society Convention*, 2004.
- [9] Stephan Herzog and Marcel Hilsamer, “Low frequency group delay equalization of vented boxes using digital correction filters,” in *Proceedings of the 16th International Conference on Digital Audio Effects*, 2014, pp. 57–64.
- [10] Elliot Kermit-Canfield and Jonathan Abel, “Signal decorrelation using perceptually informed allpass filters,” in *Proceedings of the 19th International Conference on Digital Audio Effects*, 2016, pp. 225–31.
- [11] Elliot K. Canfield-Dafilou and Jonathan S. Abel, “A group delay-based method for signal decorrelation,” in *Proceedings of the 144th Audio Engineering Society Convention*, 2018.
- [12] Elliot K. Canfield-Dafilou and Jonathan S. Abel, “Group delay-based allpass filters for abstract sound synthesis and audio effects processing,” in *Proceedings of the 21st International Conference on Digital Audio Effects*, 2018.
- [13] Vesa Välimäki, Jonathan S. Abel, and Julius O. Smith III, “Spectral delay filters,” *Journal of the Audio Engineering Society*, vol. 57, no. 7/8, pp. 521–31, 2009.
- [14] Jonathan S. Abel and Elliot K. Canfield-Dafilou, “Dispersive delay and comb filters using a modal structure,” submitted 2019.
- [15] Jonathan S. Abel, Sean Coffin, and Kyle Spratt, “A modal architecture for artificial reverberation with application to room acoustics modeling,” in *Proceedings of the 137th Audio Engineering Society Convention*, 2014.
- [16] Avery Wang and Julius O. Smith III, “On fast FIR filters implemented as tail-canceling IIR filters,” *IEEE Transactions on Signal Processing*, vol. 45, no. 6, pp. 1415–27, 1997.
- [17] Max Mathews and Julius O. Smith III, “Methods for synthesizing very high Q parametrically well behaved two pole filters,” in *Proceedings of the Stockholm Musical Acoustics Conference*, 2003.
- [18] Dana Massie, “Coefficient interpolation for the Max Mathews phasor filter,” in *Proceedings of the 133rd Audio Engineering Society Convention*, 2012.
- [19] Robert McAulay and Thomas Quatieri, “Speech analysis/synthesis based on a sinusoidal representation,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, no. 4, pp. 744–54, 1986.
- [20] Jonathan S. Abel and Kurt James Werner, “Distortion and pitch processing using a modal reverberator architecture,” in *Proceedings of the 17th International Conference on Digital Audio Effects*, 2015.