Effect Design*

Part 2: Delay-Line Modulation and Chorus

JON DATTORRO, *AES Member*

CCRMA, Stanford University, Stanford, CA, USA

The paper is a tutorial intended to serve as a reference in the field of digital audio effects in the electronic music industry for those who are new to this specialization of digital signal processing. The effects presented are those that are demanded most often, hence they will serve as a good toolbox. The algorithms chosen are of such a fundamental nature that they will find application ubiquitously and often.

4 LINEAR INTERPOLATION

4.1 Audio Applications of Interpolation: Chorus, Flange, and Vibrato Effects

The technique of delay-line interpolation is used when it is desired to delay a signal by some number of samples expressible as a whole plus some fractional part of a sample. This way, the effective delay is not discretized, thus avoiding signal discontinuities when the desired delay time is continuously swept or modulated. Delay modulation is indigenous to pitch change and pitch shift algorithms,[52] which are themselves integral to numerous other effects, such as chorus, flange, doppler, detune, harmonizer, Leslie rotating speaker emulation, or doubling. The chorus and flange effects come about when the output signal is made to be a linear sum (mix) of the original (dry) input and the dynamically delayed (wet) input signal. The chorus and flange effects are distinguished primarily by the minimum delay in their respective delay ranges.[53] Delay modulation alone (with no mix) yields vibrato when the modulation is sinusoidal.

In this section we present the topic of delay-line interpolation from the intuitive point of view of the required fractional sample delay, that is, from a time-domain viewpoint. The formal derivation, called sample rate conversion [32], [29], is traditionally a frequency-domain formulation. In a musical context, the sample-rate conversion ratio inverse (M/L in Section 6.2.3, Appendix 4) corresponds to the pitch change (or pitch shift) ratio. We synopsize the formal derivation in Section 6.2, Appendix 4, where a schematic translation of the fundamental algorithms discussed here, to the classical digital signal processing (DSP) nomenclature will be found. This should serve to bridge the two viewpoints.

The interpolation methods we seek are computationally simple and inexpensive by necessity. The interpolation algorithm may be executed many times in one sample-synchronous audio processing program. We typically cannot afford interpolation routines that consume a large percentage of the allotted execution time, since they play only a subsidiary role.

4.2 Interpolation of a Delay Line

The DSP block diagram for delay-line interpolation is shown in Fig. 22, where we find a delay line named Voice$L$ of length 2048 samples. At every sample time n the newest input sample is accepted into the left-hand side of the delay line (sample 0), while the oldest sample is discarded off the right-hand side. That action defines the delay line. The delay-line output is rarely the last sample, so the output is shown above the block as an arbitrary tap, somewhere within the body of the delay line. Because the arbitrary delay tap is moving smoothly about the tap center, it becomes necessary to interpolate

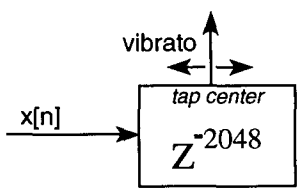


Fig. 22. Delay line Voice$L$[2048]. NOMINAL_DELAY corresponds to delay tap center.

* Manuscript received 1996 March 14; revised 1996 September 14 and 1997 June 28.

[52] Pitch change and pitch shift will be distinguished later (see Figs. 51 and 52).

[53] The minimum is less for the flange effect (Section 6). Any use of delay modulation typically entails a nominal signal delay because the modulation spans some desired range.

in real time in between the discrete samples of the delay line. The audible effect of the tap movement is vibrato.

The process called vibrato[54] is shown microscopically in Fig. 23. The desired output (vibrato) dynamically points (i.frac) to a place between two discrete samples. The index i, an integer, is defined as the current computed whole relative index into our delay line, relative to the beginning of the delay line. The integer i requires computation because we want it modulated by the low-frequency oscillator (LFO) $y[n]$, oscillating as a function of discrete time n. The integer range of i, ±CHORUS_WIDTH, is centered about the nominal tap point into the delay line, NOMINAL_DELAY the fixed positive integer tap center.

Note from Fig. 23 that because the time-varying pointer i.frac is always a nonnegative real relative index into the delay line, the elemental coefficient frac is always a nonnegative fraction.

$$i.\text{frac} \equiv i + \text{frac}$$

The time-varying nonnegative coefficients frac and (1 − frac) can have 23-bit resolution within a 24-bit processor chip because CHORUS_WIDTH (a fixed positive whole number of samples) is scaled by an LFO $y[n]$, a unity-level bipolar control signal that is typically two's complement q23 format.[55] (Sinusoidal oscillators as LFOs are discussed in Section 7.)

4.3 Linear Interpolation as Polyphase Digital Filter

Linear interpolation as applied in the chorus, flange, and vibrato effects makes small undulating changes in pitch. If we listen closely to all these effects, we will also discover a significant perceived loss of high-frequency content. For the chorus and flanger the loss we perceive is *not* attributable to the comb filtering that is constituent to those effects.[56] It is this observation which compels a frequency-domain analysis of linear interpolation.

We are not accustomed to think of linear interpolation as a frequency-domain filtering process. This is because it is such an intuitively simple time-domain algorithm that performs fractional sample delay. Yet linear interpolation is routinely used to perform sample rate conversion in contemporary sampler-type music synthesizers,[57] and we know there is a vast amount of literature that poses the sample rate conversion problem in the frequency domain.

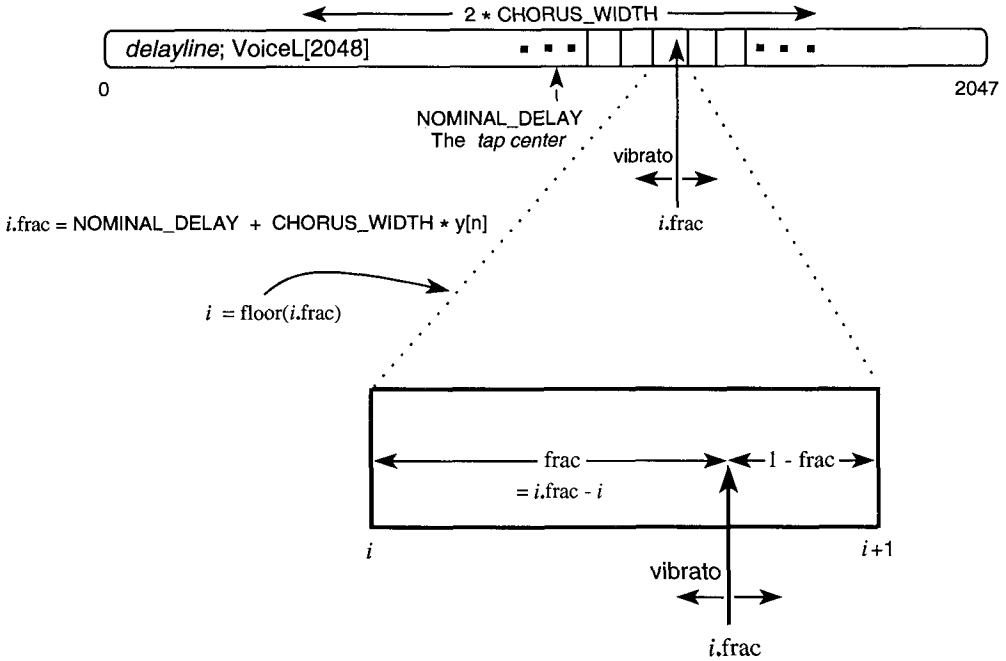
Fig. 24 shows the actual dual-input linear time-varying digital filter that implements linear interpolation. The time index n steps through discrete time in a

[54] Vibrato is most often implemented by undulating the effect we call pitch change, although it can certainly be accomplished by other means (see Figs. 51 and 52).

[55] See Part 3, Section 9.1, Appendix 7.

[56] When a signal is added to a delayed replica of itself, comb filtering results.

[57] There it most often manifests itself as a constant pitch change at playback, dependent on the key struck. (We distinguish pitch change from pitch "shift"; see Figs. 51 and 52.) The technique used to accomplish pitch change by fixed amounts is called the phase-accumulating oscillator [33]. The Ensoniq Mirage, introduced in 1984, used zeroth-order interpolation (choosing the nearest sample in time) in conjunction with eight phase-accumulating oscillators. The Mirage, one of the earliest sampler-type music synthesizers, is still in commercial use because of its characteristic sound quality due primarily to its digital encoding scheme. Its hybrid floating-point design was based on an 8-bit mantissa and an 8-bit exponent. Recorded sound samples used only the mantissa. Subsequent enveloping applied the 8-bit exponent to the analog reference on a digital-to-analog (D/A) converter. That way the signal-to-noise ratio of the original sample was not compromised when the signal was dynamically scaled. Robert Yannes.



For linear interpolation: $v[n] = \text{frac} * \text{Voice}L[i + 1] + (1 - \text{frac}) * \text{Voice}L[i]$

For all-pass interpolation: $v[n] = \text{Voice}L[i + 1] + (1 - \text{frac}) * \text{Voice}L[i] - (1 - \text{frac}) * v[n - 1]$

Fig. 23. $v[n]$ = vibrato.

sequential fashion, $0 \leq n < \infty$. The index n always refers to the current sample. We have drawn the linear interpolation circuit in a strange way to emphasize the nonsequential access of the input samples demanded by the computed index i. The unit delay z^{-1} is not terminated because the index i is time varying (it changes at each time step n), which is to say that it can take on any value within bounds.[58] From Fig. 23 we have established the parameters: $0 \leq i < 2048$. The consequence of these considerations is that $x[n - i - 1]$ is not necessarily the *old* value of $x[n - i]$.

To make the connection from the circuit in Fig. 24 to the delay line in Fig. 23, we make the analytical identification

$$x[n - i] = \text{Voice}L[i] \ .$$

This identification locates the requested sample in our delay line. It is clear that $x[n]$ always refers to Voice$L[0]$, as this is the current sample; positive i indexes older samples in our delay line.

Ideally what we want is for the actual output $v[n]$ in Fig. 24 to approximate the value of the continuous signal $x(t)$ at points in time between sample instants; that is, we would like

$$v[nT] \approx x((n - i.\text{frac})T) \qquad (39)$$

where T is the sample period. The linear interpolation circuit of Fig. 24 makes this approximation Eq. (39). It is a time-varying circuit[59] because its coefficients are a function of i.frac, as outlined in Fig. 23. The linear interpolation circuit is polyphase because whenever an output sample $v[n]$ is computed, a new pair of coefficients is fetched, yielding a different phase response from one of the filters in an ordinal set[60] [29, ch. 4.6.5, p. 166].

4.4 Movement of the *i*.frac Pointer

It is interesting to know how the pointer i.frac moves under certain conditions.[61]

4.4.1 Constant Delay

i.frac is a real constant, that is, it is not dynamic in this circumstance.

4.4.2 Undulating Pitch Change, Vibrato

It is given that i.frac varies sinusoidally. Without loss of generality we may consider vibrato as applied to an arbitrary sinusoid of constant amplitude A, phase ϕ, and radian frequency $\Omega = 2\pi f$. Using the nomenclature from Eq. (39), we write

$$x((n - i.\text{frac})T) = A \cos[\Omega(n - i.\text{frac})T + \phi]$$

where

$$i.\text{frac} = \text{NOMINAL\_DELAY} + \text{CHORUS\_WIDTH}\ y[n]$$

$$y[n] = \sin(\Omega_\varepsilon nT).$$

Here $y[n]$ is the LFO, for Ω_ε the radian rate of modulation ($2\pi f_\varepsilon$), and for the sample period T. NOMINAL_DELAY and CHORUS_WIDTH are constants expressed in whole units of samples representing, respectively, the offset and the amplitude of the LFO. So at time $n = 0$, i.frac points to NOMINAL_DELAY. The instantaneous radian frequency of $x((n - i.\text{frac})T)$ is then

$$\Omega_I = \frac{\partial\{\Omega[n - \text{NOMINAL\_DELAY} - \text{CHORUS\_WIDTH}\sin(\Omega_\varepsilon nT)]T + \phi\}}{T\,\partial n}$$

$$= \Omega[1 - \text{CHORUS\_WIDTH}\,\Omega_\varepsilon T\cos(\Omega_\varepsilon nT)]$$

and so

$$\text{pitch change ratio} = \Omega_I/\Omega$$

$$= 1 - \text{CHORUS\_WIDTH}\,\Omega_\varepsilon T\cos(\Omega_\varepsilon nT)\ ,$$

$$\text{pitch change ratio extrema} = 1 \pm \text{CHORUS\_WIDTH}\,\Omega_\varepsilon T\ .$$

[58] Strictly speaking, i.frac is a function of n, that is i.frac$[n]$ would be proper. The nonsequential nature of i demands random access of delay-line samples. So for now the unit delay in Fig. 24 is not being utilized.

[59] It is possible for a polyphase network to be time-*in*variant even when the constituting circuits are time varying. This happens when the output signal is a replica of the input signal to within a constant and/or a delay term [29].

[60] The number of possible coefficients is related to their resolution. There are $L = 2^{23}$ possible pairs of nonnegative coefficients in 24-bit two's complement, hence that many filters [34, ch. 4.3.11]. Notice that when frac = 0, which is not unusual, the linear interpolation circuit performs no filtering action. The "filter" corresponding to frac = 0 is the zeroth filter in an ordinal set consisting of L filters.

[61] One may skip to the next section without loss of continuity.

The pitch change ratio is the ratio of the new pitch Ω_I to the original pitch Ω.[62] In the present case, the pitch change is time varying, sinusoidal like the LFO, and proportional to the modulation frequency and the sample period. Note that if the LFO waveform were triangular, then the instantaneous frequency would be piecewise constant, which is unnatural.

Solving for i.frac in terms of the pitch change ratio, we determine that, in general

$$i.\text{frac} = n - \int \text{pitch change ratio}\ \partial n$$

where −NOMINAL_DELAY becomes the constant of integration.

4.4.3 Constant Pitch Change

It is given that i.frac varies linearly, that is, constant pitch change does not imply constant delay nor constant i.frac pointer.

As before, we consider an arbitrary sinusoid of constant amplitude, phase, and frequency. So using Eq. (39), we again write

$$x((n - i.\text{frac})T) = A\cos[\Omega(n - i.\text{frac})T + \phi]$$

but where

$$i.\text{frac} = \text{NOMINAL\_DELAY} + (1 - \text{pitch change ratio})\, n\ .$$

NOMINAL_DELAY is constant, expressed in whole units of samples, representing the pointer offset. At time $n = 0$, i.frac points to NOMINAL_DELAY. The pitch change ratio is also assumed constant. The instantaneous radian frequency of $x((n - i.\text{frac})T)$ is then

$$\Omega_I = \frac{\partial[\Omega(-\text{NOMINAL\_DELAY} + \text{pitch change ratio} * n)T + \phi]}{T\,\partial n}$$

$$= \Omega * \text{pitch change ratio}\ .$$

This verifies that the pitch change is indeed constant when i.frac varies linearly. Unfortunately i.frac will eventually pass one or the other delay-line boundary, so this technique cannot be used indefinitely.

4.5 High-Frequency Loss of Linear Interpolator

We need to delve further into the connection between linear interpolation as a time-domain process and the filtering anomaly mentioned, which we perceive in the chorus, flange, and vibrato effects. To facilitate the analytical portion of the investigation, we will approximate the actual circuit used (Fig. 24), substituting instead the polyphase filter circuit of Fig. 25 into the analysis.[63] The unit delay z^{-1} of Fig. 24 is now terminated in Fig. 25. We thereby ignore the second input in the circuit of Fig. 24 when making the analytical approximation because the two inputs there are separated by one sample in the delay line. To justify that for the analysis, we need not assume constant i because we will view the instantaneous transfer function[64] of this nonrecursive circuit in Fig. 25.

Because of the constant interrelationship of the time-varying coefficients, the analytical approximation has an instantaneous frequency response that traverses a range from an all-pass transfer function $[1, z^{-1})$ at either extreme of the coefficients, to an averaging transfer function $(1 + z^{-1})/2$ in the middle. This large set of transfers represent the frequency responses of the polyphase filters. Fig. 26 shows several of these magnitude responses, corresponding to 20 equispaced values of frac, that is, frac = 0→0.95 spaced by 0.05.

It is this set of magnitude responses (Fig. 26) that we hear in our effects.[65] In some musical applications, flutter is clearly audible and objectionable. In other applications, a veil seems to have been placed over the sound source.[66]

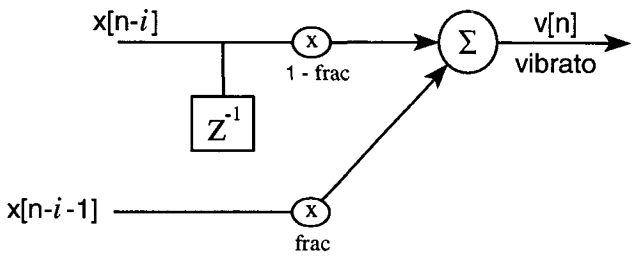


Fig. 24. Linear interpolation circuit.

[62] The pitch change ratio is defined as a ratio of strictly positive integers M/L in Section 6.2.3, Appendix 4. Where the pitch change ratio becomes a function of time, M becomes a function of n. In the present circumstance, were the pitch change ratio to go negative, that would mean that samples were being read from the delay line backward in time.

[63] The time-varying approximation to linear interpolation (Fig. 25) is only used for analysis; it is not used in the actual implementation.

[64] We freeze time and then determine the transfer function at that moment. This analytical device is justifiable if i.frac changes slowly in time.

[65] Each curve except for the extreme curves is duplicated since there are two sets of coefficients corresponding to each curve in the case of linear interpolation. For example, $(0.25 + 0.75z^{-1})$ and $(0.75 + 0.25z^{-1})$.

[66] For many of the audio effects in which linear interpolation is used, i.frac changes slowly enough that the concomitant dynamic filtering is objectionably audible. For example, near-unity pitch change ratio for sounds having much high-frequency content produces audible flutter. When i.frac changes faster than our ability to perceive the flutter, our hearing system integrates the response of many of these polyphase filters, each operating at a subsample rate ($\leq F_s$). In this circumstance one must use the more rigorous mathematical description [see Eq. (54)] to explain the filtering anomaly that remains audible, like the veil, but is no longer dynamic.

4.6 Fractional Sample Delay of Linear Interpolator

Fig. 27 shows the delay response as a function of frequency introduced by the polyphase filters corresponding to Fig. 26. We may rightfully deduce that the polyphase filters are responsible for only the fractional portion of the desired sample delay determined by i.frac. This fractional sample delay is figured as the negative of the phase response divided by the normalized radian frequency, namely, the phase delay,

$$\hat{\tau} = -\frac{\arg[V'(e^{j\omega})/X(e^{j\omega})]}{\omega}.$$

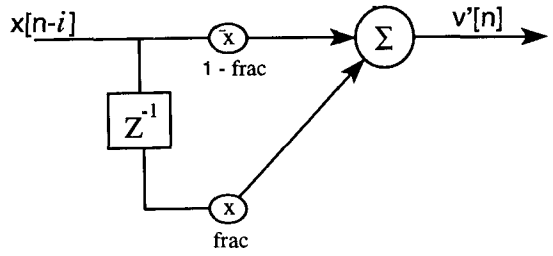


Fig. 25. Analytical approximation of linear interpolation.

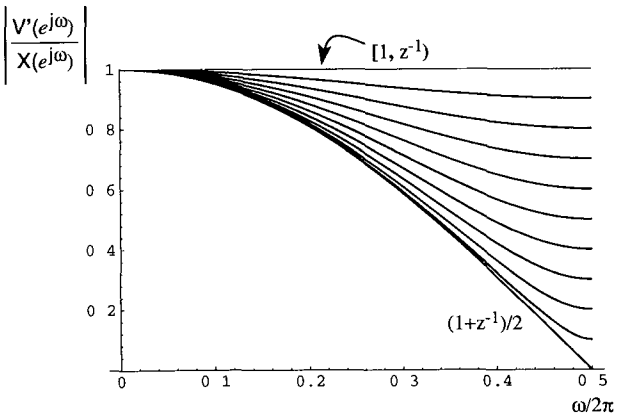


Fig. 26. Magnitude responses of linear interpolation polyphase filters; $L = 20$.

This actual signal delay is with reference to a steady-state sinusoid input. From Fig. 23 we can see that the *desired* fractional sample delay τ is frac, which is constant when time is frozen and independent of frequency.

The delay responses of the polyphase filters are as important as their magnitude responses. When the coefficient frac calls, say, for a quarter-sample delay, we would like a constant quarter-sample delay for all frequencies. The only transfer function of linear interpolation that meets this criterion perfectly is the one in the middle, $(1 + z^{-1})/2$; that is, it is the only transfer, aside from the trivial case of frac = 0, which we will see having linear phase, hence a constant delay (half-sample in this case). Ironically, it is observed from Figs. 26 and 27 that when the filtering is at its worst, the delay is ideal.

Fig. 28 shows the average delay (over frequency) of each individual polyphase filter for the linear interpolator,

$$\bar{\hat{\tau}} = \frac{-1}{\pi}\int_0^{\pi} \frac{\arg[V'(e^{j\omega})/X(e^{j\omega})]}{\omega}\,d\omega.$$

This average actual delay is calculated for every point on the curve in the graph, corresponding to a different value of the coefficient frac, the desired fractional sample delay. There are $L = 2^{23}$ different polyphase filters because that is how many different nonnegative coefficients there are in 24-bit two's complement [34, ch. 4.3.11], q23 format. The straight line is the desired average.

4.6.1 Interpolation Distortion

Notice that in general the fractional sample delay for each individual polyphase filter in Fig. 27 is fairly constant in the low-frequency region. If we regard only dc in Fig. 27, we make the further observation that the actual fractional sample delay perfectly tracks the polyphase filter coefficient sample frac, as desired (see Fig. 23). This property is a major determinant in the successful performance of the linear interpolator from the standpoint of total harmonic distortion + noise (*THD+N*)

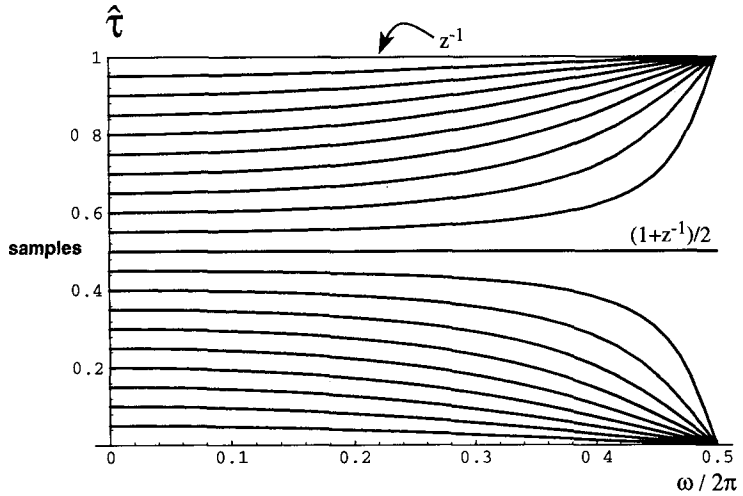


Fig. 27. Delays of linear interpolation polyphase filters. Extreme polyphase filter z^{-1} is not used.

of low-frequency signals subject to pitch change.[67]

To understand why, recall from Section 4.4 that constant or undulating pitch change requires a time-varying delay of the input signal. If the time-varying filter provides a succession of delays that are in error, phase distortion will be introduced into the pitch-changed output signal. From Fig. 27 we learn that the only frequency for which there is no delay error is dc. This means in general that phase distortion will be introduced into the linear interpolation of any sinusoid having nonzero frequency. The distortion worsens as the frequency becomes higher.

Nearly the same observations can be made regarding amplitude distortion, as revealed by Fig. 26, where we learn that the only frequency for which there is no amplitude error is dc. In summary, there are at least two components constituting the distortion introduced by linear interpolation—phase and amplitude. In what follows, we will attempt to eliminate the amplitude distortion.

5 ALL-PASS INTERPOLATION

The implementation cost versus performance of linear interpolation is difficult to beat, especially as the sample rate is increased. It does have significant drawbacks, however, which motivate us to look for better methods.

The following are drawbacks to the use of linear interpolation:

1a) *Amplitude Distortion*: Linear interpolation is a low-pass process having a dynamic zero at the Nyquist frequency (Fig. 26), creating muffled (veil-like) sounds and unaccounted damping in signal paths employing it.

1b) *Amplitude Modulation*: The dynamic zero of the polyphase digital filters of linear interpolation introduces audible flutter, which is quite objectionable near unity pitch change ratio (M/L) for sounds having much high-frequency content.

2) *Phase Distortion*: This arises due to the nonconstant delay response of each polyphase filter (Fig. 27).

3) *Aliasing*: An exceptional amount of it occurs and is worst for large pitch change upward, corresponding to the case of sample-rate reduction (decimation).[68] In the other direction we can link aliasing to phase distortion (see Section 5.3).

We could eliminate the amplitude artifacts 1a) and 1b) if the polyphase filters were all pass. So we need to know whether it is possible to make all the individual (frozen) polyphase filters have all-pass transfers while still performing interpolation. The answer is in the affirmative [29, ch. 4.6.5, p. 166], [35], and we explore nearly all-pass polyphase filters in this section.[69]

Engineers within the audio industry [36],[70] [37] report alternative interpolation strategies.[71] We recommend

[67] ***THD+N*** is inherent to any interpolation process. It is a measure of signal purity, which aggregates everything that is not signal, and then relates that to some purified reference, which is ideally the signal itself—more apt, $S/(THD+N)$.

[68] This last problem comes about because the linear interpolation prototype filter has one-sided bandwidth π/L. The same is true for all-pass interpolation (see Figs. 47–49). The bandwidth is not a design parameter, but falls out as a result of these techniques. When the downsampling factor M exceeds the upsampling factor L, rather upon decimation, aliasing is traditionally tolerated for these two techniques (see Section 6.2.4, Appendix 4), because it is unavoidable unless the input signal is pre-filtered.

[69] The reason that the polyphase filters can each have an ideal all-pass transfer function is discussed in Section 6.2.4, Appendix 4. The formal frequency-domain formulation of interpolation derives the polyphase filters from what is called the prototype interpolation filter. The prototype filter (Section 6.2.4) must not be all pass, by definition, but no such restriction is placed on the individual polyphase filter. In fact, for the idealized formulation of interpolation by a rational factor, each time-invariant polyphase filter is exactly all pass and linear phase [29, ch. 4, pp. 168, 109, 124], [34, ch. 4 2.2]. But the polyphase filters of linear interpolation are clearly not all pass, as illustrated in Fig. 26. The corresponding prototype filter in Fig. 47 has a discrete triangular finite impulse response, illustrated in Fig. 44 [14, ch. 3.6.2, p. 109].

[70] Pitch change algorithms are actually the topic despite the title.

[71] The E-mu Proteus sampling music synthesizer (1989) and its relatives all employ seventh-order interpolation polynomials. They are not exactly "Lagrange" though. They use a technique in which a Remez exchange is applied to an "ideal" filter response similar to that of Lagrange, but having lower maxima in the stopband. This gives the deep notches advantageous in the Lagrange approach, but also the superior stopband rejection of a sinc-based design. For more information, see the U.S. patent on the fundamental E-mu G-chip interpolator; no. 5,111,727. David Rossum.

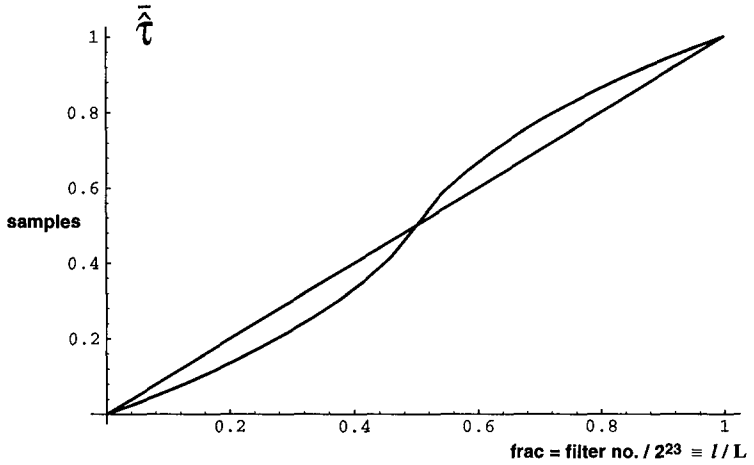


Fig. 28. Average delay of each linear interpolation polyphase filter.

Lagrange interpolation when higher order finite impulse response (FIR) filters are a viable option [32], [34], [38], [39]. Lagrange interpolation is an analytical extension to linear interpolation.[72] But nonrecursive techniques such as these have high computational cost. Nonetheless, FIR filters dominate contemporary sample-rate conversion practice because from the point of view of internal truncation noise, it is difficult to mess up an FIR implementation [40]–[42]. Also, FIR filters offer linear phase.[73]

Recursive polyphase filters have not been popular because they are not linear phase, in general.[74] The practice of recursive digital filtering requires an understanding of fixed-point arithmetic, truncation error recirculation [12], and transient phenomena.

5.1 All-Pass Interpolation as Polyphase Digital Filter

We present here the simple recursive technique of all-pass interpolation, which is useful primarily for microtonal changes in pitch (less than plus or minus one semitone). Linear interpolation will outperform it from the standpoint of ***THD+N*** (Section 5.4). Otherwise all-pass interpolation minimizes the drawbacks of linear interpolation in this microtonal region and makes the interpolation sound analog.

Fig. 29, a modification of Fig. 24, shows the actual circuit used to implement all-pass interpolation [43]. Our application of Fig. 29 uses time-varying filter coefficients. The formal derivation of the classical polyphase all-pass interpolation network [29], [35] requires as many recursive memory elements as there are coefficients ($L = 2^{23}$ in 24-bit two's complement).[75] This is one reason why this simpler circuit in Fig. 29, employing only one recursive element with a time-varying coefficient, only performs well (in terms of ***THD+N***) for small pitch changes. Making the same analytical approximation as before, we can see that connection of the node at $x[n - i - 1]$ instead to the unit delay would make the instantaneous transfer function of the resulting polyphase filter all pass. This means that the interpolation circuit in Fig. 29 has a frequency response that is approximately all pass; $|V(e^{j\omega})/X(e^{j\omega})| \approx 1$.

5.2 Fractional Sample Delay of All-Pass Interpolator

The time-varying coefficient (1 − frac) in Fig. 29 is easily derived from $i.frac$, as in Fig. 23. We find that, subjectively, the circuit of Fig. 29 sounds quite smooth in musical applications. But analytically speaking, the actual signal delay introduced by the analytical approximation does not track frac as well as it does for linear interpolation.

Fig. 30 corresponds to Fig. 27. Each curve in Fig. 30 corresponds to one of 20 equispaced values of frac, that is, frac = 0→0.95, spaced by 0.05, as before. Notice how the delay between filters is spaced unevenly, however, which is especially noticeable at dc. But for each individual filter the delay is still fairly constant in the low-frequency region. Evidently, the average delay (over frequency) of each polyphase filter, shown in Fig. 31, is better than that for linear interpolation (compare to Fig. 28).

5.2.1 *All-Pass Interpolator Coefficient Warping*

It might be reasonable to expect, mathematically speaking, that we could force the delay of each polyphase filter to be equal to any desired value at one partic-

[72] That is, a higher order polynomial curve fit using more signal values, and which is maximally flat in the frequency domain while suppressing ripple in the time domain. The two-point Lagrange interpolator is equivalent to linear interpolation.

[73] Generally speaking, a linear-phase prototype interpolation filter does not guarantee linear-phase polyphase filters, and vice versa. Linear interpolation, for example, is an eminent case of an FIR prototype that is linear phase (having a symmetrical triangular impulse response of length $2L$ spanning two original samples by design [14, ch. 3.6.2, p. 109], [34]) but whose polyphase filters are *not*. Yet if a linear-phase prototype is ideally band-limited to π/L, for L a rate conversion factor, then all its L polyphase filters will remain exactly linear phase [37, p. 545], [29, ch. 4.6.5, p. 168], [14, ch. 5.7]. (See Section 6.2, Appendix 4.) Crochiere and Rabiner [34, ch. 4.3.6–4.3.10] give explicit general design procedures for simultaneously linear-phase FIR polyphase and prototype interpolation filters.

[74] Renfors and Saramäki's infinite impulse response (IIR) design offers nearly linear-phase recursive polyphase and prototype interpolation filters [35].

[75] In the classical polyphase network, all the polyphase filter coefficients are fixed. Each recursive memory element resides in a structure like that in Fig. 29, having the feedforward delay element connected. Our simulations have shown that to implement the classical polyphase all-pass interpolation network, as few as $L = 2^8$ recursive elements work quite well to make constant or sweeping pitch change over a *large* range (see Section 6.2.10, Appendix 4). On the other hand, even the most typical implementation of linear interpolation corresponds perfectly to its classical polyphase counterpart simply because of the lack of required long-term memory [34, ch. 3.3.2, p. 81; ch. 4.3 11]. Hence the size of L is constrained for linear interpolation only by the coefficient resolution. The higher the resolution, the higher becomes L.

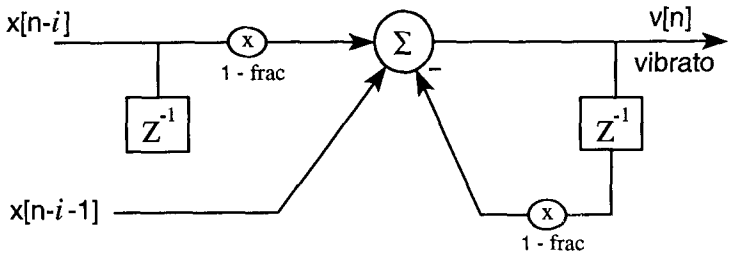


Fig. 29. All-pass interpolation circuit.

ular frequency by appropriately altering the formula for the polyphase filter coefficients. This would be beneficial if we were concerned with ***THD+N*** performance in a particular frequency region. Indeed, the alteration that precisely warps the all-pass circuit coefficients in Fig. 29 is the substitution [44, p. 178]

$$1 - \text{frac} \rightarrow \frac{\sin[(\omega/2)(1 - \tau)]}{\sin[(\omega/2)(1 + \tau)]}$$

where τ is the desired fractional sample delay (in units of samples) at the desired normalized radian frequency ω.

We can eliminate the dependence of the warp equation on the signal frequency ω when we are primarily concerned with ***THD+N*** performance at low frequencies (ω near 0). For then the coefficient warp equation simplifies to [24]

$$1 - \text{frac} \rightarrow \approx \frac{1 - \tau}{1 + \tau} = \frac{1}{3} + \sum_{p=1}^{\infty} \frac{(-2)^{p+2}}{3^{p+1}} \left(\tau - \frac{1}{2}\right)^p . \quad (40)$$

This simplified warp equation is a reasonable substitution for audio signals that consist predominantly of low frequencies with respect to the sample rate.[76] From Fig. 23 we can see that the desired fractional sample delay is frac; we recall the identification

$$\tau = \text{frac} \quad (41)$$

where frac is the elemental polyphase filter coefficient. Thus we arrive at Fig. 32, which shows the actual time-varying circuit used to implement warped all-pass interpolation using the simplified warp equation Eq. (40) in place of the all-pass coefficients in Fig. 29.

Fig. 33, like Fig. 30, also corresponds to Fig. 27 and is plotted with the same set of values for frac. It demonstrates the even distribution of delay across the polyphase filters at low frequency using the simplified warp equation Eq. (40) as in Fig. 32. This closer tracking between frac and actual signal delay will help improve somewhat the ***THD+N*** performance at low frequencies by diminishing phase distortion there.

But the improvement in the delay distribution in the low-frequency region causes the average delay, shown in Fig. 34, to suffer (compare to Fig. 31). We find, analytically, that for all-pass interpolation of low-frequency sinusoids up to a few kilohertz, use of the simplified coefficient-warp equation Eq. (40) is desirable from

[76] The series form of the substitution from the simplified warp equation [Eq. (40)] will become handy when we implement this new polyphase filter coefficient.

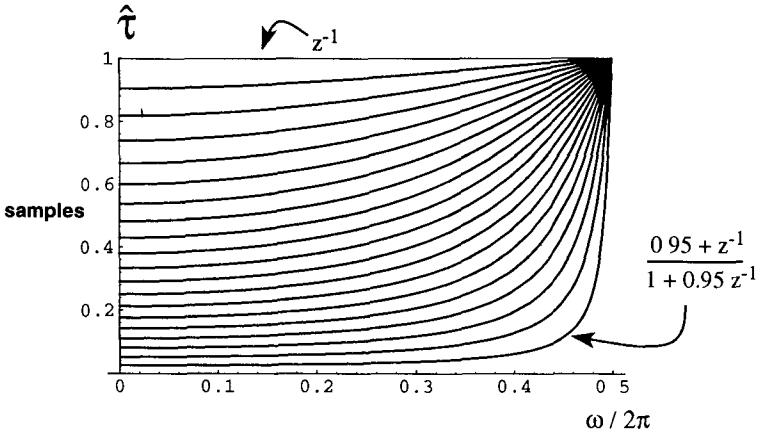


Fig. 30. Delays of all-pass interpolation polyphase filters Extreme polyphase filter z^{-1} is not used.

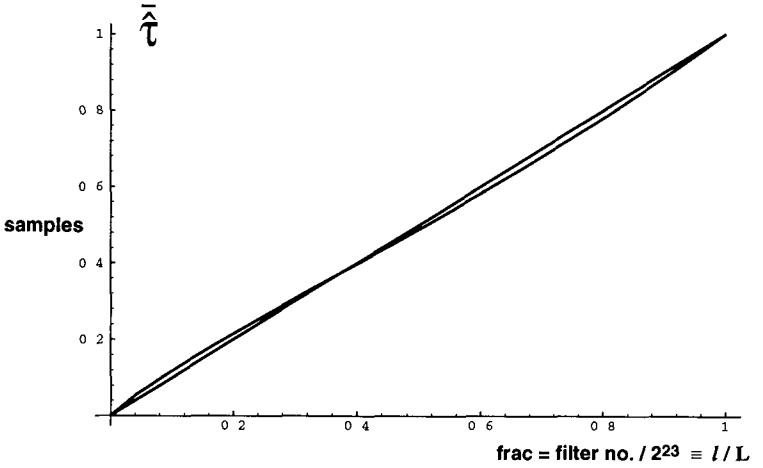


Fig. 31. Average delay of each all-pass interpolation polyphase filter.

a ***THD+N*** standpoint. Within an actual implementation, several computed terms from the Taylor series expansion Eq. (40) of the simplified warp equation (or a table lookup routine) would easily map frac to the warped coefficients of Fig. 32. So it is worth doing (Section 5.4).

Notwithstanding the foregoing, for some of our musical applications we have not found it absolutely necessary to implement the all-pass interpolation circuit using coefficients different from those shown in Fig. 29. In other words, we have not found coefficient warping as illustrated in Fig. 32 absolutely necessary for all musical purposes, because in some musical contexts the induced phase distortion is not subjectively objectionable. That is not to say that the distortion is never objectionable. The effect designer must make the decision to incur the extra computational expense based upon the intended audience.

5.3 Distortion Analysis

Fig. 35 shows estimates of ***THD+N*** of a 16-bit 401 Hz sinusoid, sampled at 44.1 kHz, for various constant pitch changes spanning plus or minus one semitone. The C program simulation that produced the ***THD+N*** estimates emulates fixed-point arithmetic such as would be found within a contemporary 24/48-bit DSP chip [1]. Thus the signal paths are 24 bit in width and the accumulations are double precision. Truncation is post-accumulation; memory is presumed 24 bit wide. The sinusoid $x[n]$ is passed through the interpolation circuits of Fig. 24 (linear) and Fig. 32 (warped all pass). Then

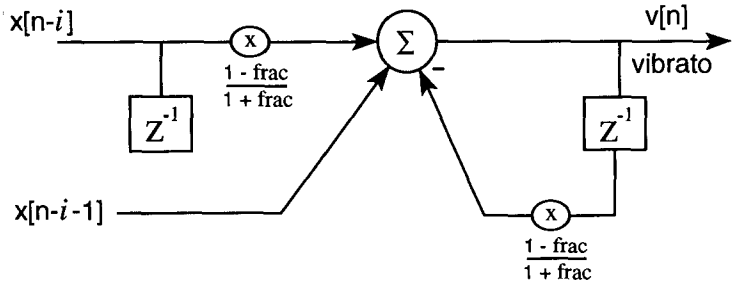


Fig. 32. Warped all-pass interpolation circuit

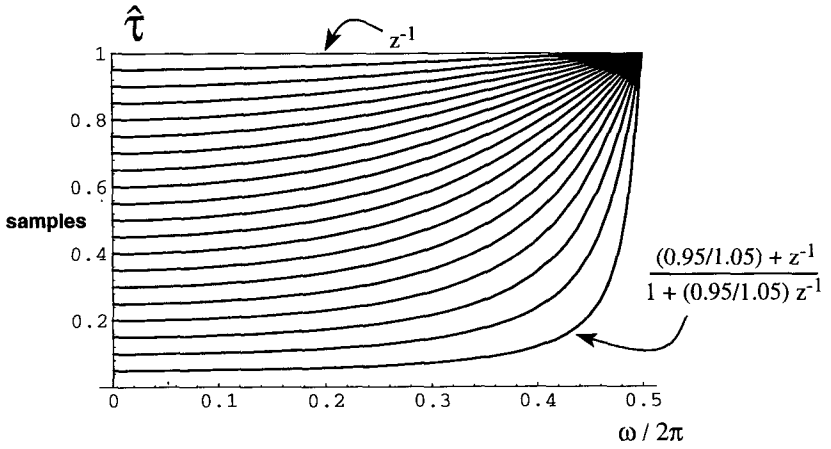


Fig. 33. Delays of warped all-pass interpolation polyphase filters. Extreme polyphase filter z^{-1} is not used.

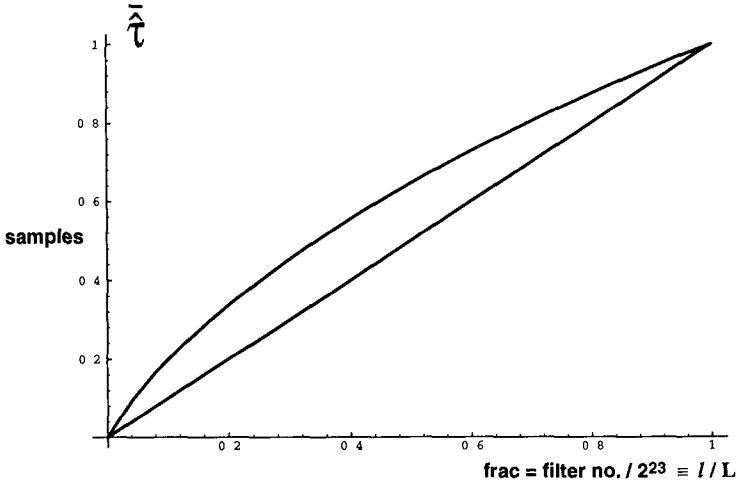


Fig. 34. Average delay of each warped all-pass interpolation polyphase filter.

a mean-square error is calculated between the actual circuit output and the ideal (floating-point) desired output. The desired output is calculated, that is, an ideal input is *not* passed through an ideal circuit. That is why the plots in Fig. 35 bottom out at about −96 dB, the presumed distortion of the 16-bit sinusoid input. The mean-square error forms a ratio with the desired output signal power to calculate decibels.

The estimates for all-pass interpolation in Fig. 35(b) were made using the simplified warp equation Eq. (40) and 16-bit polyphase filter coefficients, two's complement. Without Eq. (40), the ***THD+N*** curve in Fig. 35(b) would be relatively flat [like that in Fig. 35(a)] and average about −44 dB, dominated by harmonic distortion. The polyphase filter coefficients used in making the ***THD+N*** estimates for linear interpolation in Fig. 35(a) are also 16 bit in width, two's complement.[77]

The predominant artifact exposed in Fig. 35(a) is harmonic distortion due to two facts: 1) the two-tap FIR (Fig. 24) magnitude response is modulated by the time-varying filter coefficients (Fig. 26), and 2) the actual time-varying delay (Fig. 27) of the two-tap FIR deviates from that desired. Thus we have amplitude and phase distortion, as discussed previously.

The use of (warped) all-pass interpolation (Fig. 32) now warrants the consideration of transient phenomena as well as truncation error recirculation. Through experimentation via the C program simulation, we discover that the predominant artifact exposed in Fig. 35(b) is distortion due to transients arising from the time-varying filter coefficients. Although the filter coefficients are updated at the sample rate, the nature of the pitch change algorithm (as outlined in Fig. 23) demands a disjunct sequence of desired fractional-sample delay. Because the filter states (memory elements) are ignored when the coefficients change, the recursive filter responds to the abrupt change in coefficient just as it would to an abrupt change in the input signal; that is, the filter's natural response is elicited.

Now that we have graduated to a recursive interpolation filter topology, signal truncation errors will recirculate and become amplified by the single pole at Nyquist; that is, we have significant truncation noise. The simulation reveals that this particular distortion, in the form of deterministic additive noise [12], is swamped by the transient distortion. (See Section 6.2.10, Appendix 4.)

In summary, all-pass interpolation has minimized amplitude distortion while phase distortion (Fig. 33) persists.[78] The predominant distortion introduced by all-pass interpolation, however, is due to the transient response induced

[77] We recommend a minimum of 9 bit, two's complement, for linear interpolation.

[78] We learn from Appendix 4 (Section 6.2) that when the polyphase filters are all pass and the fractional sample delay for each is ideally and appropriately constant over frequency, then aliasing of signals that are confined to some low-frequency region $\sigma < \pi$ becomes impossible for σ/π less than the pitch change ratio inverse. That is because the prototype interpolation filter becomes ideal and perfectly band-limited to the Nyquist(L) frequency [Eq. (47)]. Hence it is unlikely for a replication of it to leak into the baseband. We only mention this here because it establishes a direct tie between aliasing and phase distortion. When phase distortion exists, the prototype filter cannot be ideal, that is, aliasing of low-frequency signals is a consequence of phase distortion. That is why we have not considered aliasing in this analysis.

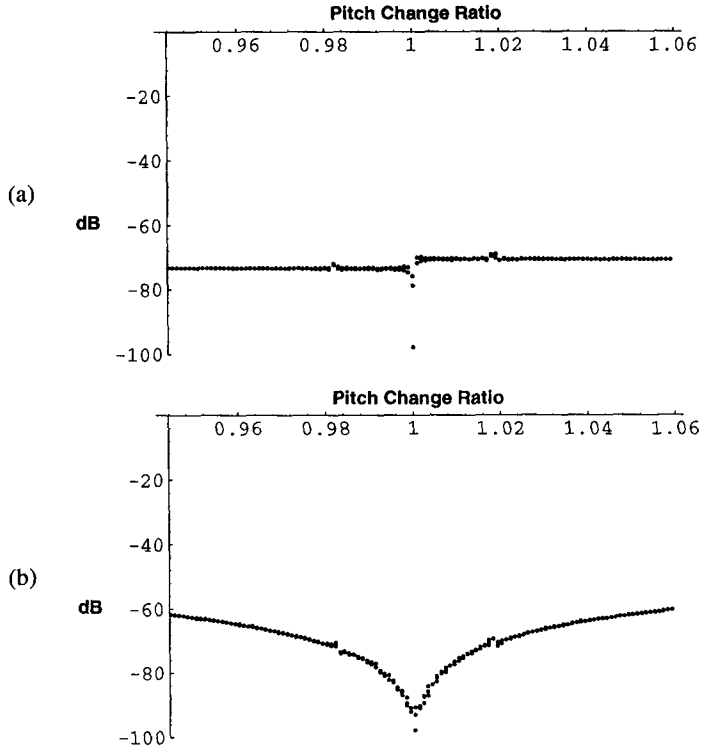


Fig. 35. Estimated ***THD+N*** of pitch-changed sinusoid. (a) Due to linear interpolation. (b) Due to warped all-pass interpolation.

by the time-varying filter coefficients in the recursive interpolation circuit. This type of distortion is not present for the linear interpolator. It is that transient distortion which bottlenecks the useful transposition range of the all-pass interpolator to about plus or minus one semitone. There is a way to overcome this particular problem (which is discussed in Section 6.2.10, Appendix 4), but the solution adds more computational expense.

5.4 Implementations of All-Pass Interpolation

Now we leave the realm of simulation and estimation to visit the reality of actual implementation and measurement. Establishing the sinusoidal LFO frequency = 0.1 Hz in a real-time 24/48-bit DSP hardware development system, we put a 24-bit 400-Hz sinusoid through a delay line, perform delay modulation, then send the 16-bit vibrato output (as in Fig. 22) to an Audio Precision signal analyzer. All polyphase filter coefficients are now 24 bit, the sample rate is 44.1 kHz to within 0.001% accuracy, and truncation is post-accumulation.

We require that the implementation of the interpolation circuit (in Fig. 29) prevent prolonged Nyquist ($F_s/2$) oscillation that comes about when frac = 0. We can accomplish this by forcing that circuit to interpolate by a constant fraction of a sample at all times, say, 1/256 sample. This technique introduces a constant fractional offset into the equation for i.frac in Fig. 23 (not shown there) and has little deleterious audible consequence,

Measured distortion of linear interpolation: $-78 \rightarrow -88$ dB

Measured distortion of all-pass interpolation: $-53 \rightarrow -59$ dB .

These ***THD+N*** measurements are time varying because the sinusoid frequency is slowly undulating due to the vibrato effect. That is why they are indicated as a range.

Next we observe the impact of coefficient warping, as shown in Fig. 32, to improve the processing ***THD+N***. Practical all-pass interpolator code uses only the first five terms of the Taylor series expansion [Eq. (40)] to map frac into (1 − frac)/(1 + frac). The code would compute the warped coefficient via the numerical approximation,[79]

$$\frac{1-\text{frac}}{1+\text{frac}} \approx \frac{1}{3} + \left(\text{frac} - \frac{1}{2}\right)\left[-\frac{8}{9} + \left(\text{frac} - \frac{1}{2}\right)\left\{\frac{16}{27} + \left(\text{frac} - \frac{1}{2}\right)\left[-\frac{32}{81} + \left(\text{frac} - \frac{1}{2}\right)\left(\frac{64}{243}\right)\right]\right\}\right] .$$

Since this map is only approximate, it is no longer necessary to prevent prolonged Nyquist oscillation as before. We measure an improvement of 26 dB in ***THD+N*** for all-pass interpolation, attributable to the coefficient warp. Linear interpolation exceeds this particular ***THD+N*** performance, but by only a few decibels,

Measured distortion of linear interpolation: $-78 \rightarrow -88$ dB

Measured distortion of warped all-pass interpolation: $-77 \rightarrow -85$ dB .

These measurements apparently lay in the groove of the estimates of Fig. 35.

5.5 Conclusions

Having reached parity between the two processes in terms of ***THD+N***, we would preferentially choose warped all-pass interpolation because it minimizes the drawbacks of linear interpolation, stated at the outset of this section, when used in a microtonal region. If the added computation to warp the coefficients is not affordable, then (nonwarped) all-pass interpolation is a viable alternative because it sounds better than linear interpolation in some musical contexts. Its computational complexity is only slightly greater than that of linear interpolation, and it minimizes amplitude distortion.

In Appendix 4 (Section 6.2) we present the classical polyphase warped all-pass interpolator as a means of pitch change over a much larger range and to a higher degree of accuracy in terms of ***THD+N***. It is primarily of theoretical interest, being computationally expensive by today's standards.

Are we on the right track? This idea of using all-pass filters to perform interpolation is somewhat foreign. But as we discover in Appendix 4, the linear-phase all-pass filter is the ideal polyphase filter in the classical formulation of interpolation [Eq. (39)] by a rational factor. So the answer is, yes indeed.

5.6 Theoretical Extensions

Laakso et al. [39] broaden the scope of this warped all-pass approach to interpolation. They provide formulas for all-pass polyphase filters of higher order that possess a more constant delay versus frequency, thereby providing a prototype interpolation filter (see Sections 6.2.4 and 6.2.7, Appendix 4) that has higher stopband rejection. This progressive linearization of the phase response as filter order is increased, however, does not proceed as quickly as we might like. Välimäki et al. [45] deal with transient phenomena.

6 WHITE CHORUS EFFECT

Any complete discussion of the chorus effect must consider its relative, the flanger. The intended goal of

[79] Beware of intermediate overflow in the calculation of this mapping.

chorusing is to emulate the independence of multiple like-voices playing in unison. But the goal of flanging is to jolt the ear's time-correlation mechanism by juxtaposition of the input signal with a replica dynamically delayed by an amount that is within the integration time constant of the hearing system.[80] We must first note that there is a very strong bond between the design of the chorus effect and of the sonically radical flange effect. What has gestated into the industry standard for both members of this species condenses simply to a sum of the original input signal with a dynamically delayed replica, namely, two voices. In either effect, the replica delay is modulating and never static. So the consequence of summing the two signals is to introduce a comb of moving troughs into the input signal spectrum [46]. In the case of flanging, that is the desired result. The deeper and more selective the troughs are, the better.[81] In the case of chorusing, the troughs are undesirable and an effort is made to globally limit their depth by summing unequal amounts of original signal and delayed replica. But the primary distinguishing design feature of the chorus is that the *minimum* of its modulating delay time is greater than that for the flange effect. This is to avoid flanging by the chorus, which becomes subjectively more pronounced for small delays. Indeed, the best flangers can sweep the delay all the way to absolute zero, that is, to no delay.

In Fig. 36 we present a modification to the industry standard chorus effect, which attempts to compensate for the spectral aberration caused by the many troughs in the feedforward sum. The modification is the introduction of a negative feedback path into the delay line, whose tap point is separate from that of the feedforward path but fixed at the center of the modulating delay in the feedforward path. Hence, the chorus circuit approximates an all-pass filter when the interrelationship of the coefficients is correct, and when the changing delay in the feedforward path is proximal with the fixed delay in the feedback path. We prefer not to feed back a modulating signal because the modulation induces pitch change. Feeding back a pitch changed signal produces more pitch change and becomes objectionable quickly for either the chorus or the flanger. Feedback is used by the flanger in an inverse sense to enhance the perceived depth of the troughs by heightening the crests in between, that is, the same circuit can be used for both flange and chorus effects.

The knob (coefficient) settings in Table 6 are given as typical parameter values for the circuit in Fig. 36. All the listed effects can be implemented using the same circuit by setting the knobs as indicated, and by choosing an appropriate tap center in accordance with Table 7. The particular knob settings given for the white chorus optimize the circuit's approximation to an all-pass response when the feedforward and feedback delays are

[80] The familiar thunder of a jet aircraft often reaches our ears by combination in air of the direct and the reflected engine backwash. As the aircraft changes position, the reflection time changes, and Doppler pitch change due to the aircraft's recession is introduced into both paths. The roar is more interesting as the reflection time is swept. That introduces more Doppler into the reflected path.

[81] The flange effect gets its name from a studio technique that sums two synchronous magnetic-tape-recorder signals playing identical material. The recording engineer places the thumb on one tape flange to bring the two recorders slightly out of synchronization, thus creating the effect. In the 1970s the flange effect was emulated by analog phase-shifting networks consisting of a cascade of all-pass filters having time-varying elements. Implemented in this manner, the problem of delaying a signal by brute force was overcome. These devices were called phasers and remain popular because the spectral troughs are not harmonically spaced, in general. While second-order all-pass filter sections in the cascade offer more control over trough frequency and selectivity [46], the phaser is well emulated in DSP using only first-order sections [47], [48] (which may be quieter in terms of truncation noise performance). The spectral trough frequencies are harmonically stretched in the first-order case. In any case, global feedback enhances the effect, affecting the perceived trough depth.

Table 6. Knob settings.

Effect	Blend	Feedforward	Feedback
Vibrato	0.0	1.0	0.0
Flanger	0.7071	0.7071	−0 7071
Industry standard chorus	1.0	0.7071	0.0
White chorus	0.7071	1.0	0.7071
Doubling	0.7071	0.7071	0.0
Echo[82]	1.0	≤1.0	<1.0

[82] Echo is accomplished with either the feedforward or the feedback knob set to zero. First-order low-pass filters are placed in those paths to simulate acoustical absorption. The filter cutoff frequency becomes another knob. The delay modulation is turned off.

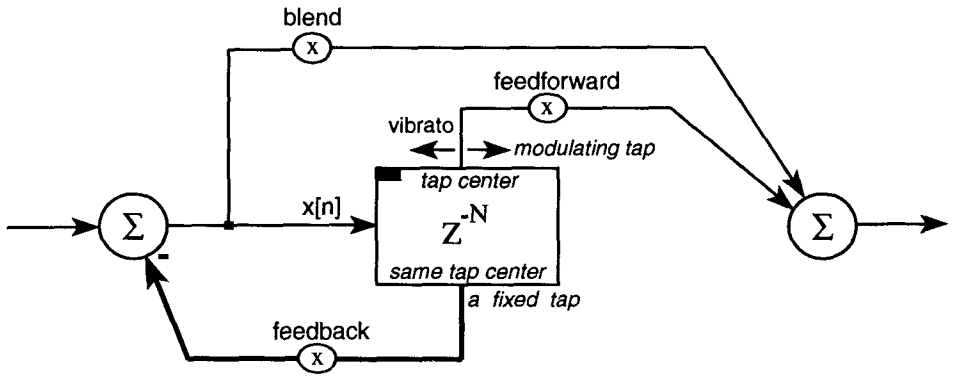


Fig. 36. Industry standard chorus effect circuit with feedback. Strong flange zone is indicated; leftmost (first) 1 ms of delay line.

proximal. To maintain that optimization, the feedforward coefficient must remain 1.0 while the blend and feedback coefficients remain equal,

$$H_{chorus}(z) = \frac{\text{blend} + \text{feedforward}\, z^{-i}}{1 + \text{feedback}\, z^{-\text{same tap center}}}.$$

Under the stated conditions on the coefficients, $|H_{chorus}(e^{j\omega})| = 1$ when i = same tap center. When the circuit in Fig. 36 performs flanging, the blend and feedforward coefficients must be equal for maximum trough depth. The maximum magnitude of feedback is 0.9999999 (q23) for stability of all the effects.

For the white chorus effect introduced here, an interesting delay tap center (NOMINAL_DELAY; Fig. 23) would be 400 samples at F_s = 44.1 kHz, whereas a musically useful peak delay excursion (CHORUS_WIDTH) of the modulation about the tap center would be approximately 350 samples. A typical rate of modulation would be about 0.15 Hz.

Table 7. Approximate effect delay range in milliseconds.

Effect	Onset	Nominal	Range End
Vibrato[83]	0	Minimal	5
Flange	0	1	10
Chorus	1	5	30
Doubling	10	20	100
Echo	50	80	∞

6.1 Chorus Effect Design

The circuit in Fig. 36 produces a brilliant tone quality, having pleasing movement with a little spatial ambience. Used singly and without feedback, this circuit's effect is called *doubling*[84] when the delay tap center reaches about 20 ms.[85]

The circuit in Fig. 36 is so simple that it is economical to use two of them. Each would process a separate channel of a stereo input signal. In that design configuration, a quadrature LFO typically provides the delay modulator so that each chorus circuit operates having 90° relative phase displacement of its modulation.[86] Each circuit output would be routed to a separate channel in a stereo pair. This quadrature strategy dynamically alters the stereo placement of the output signal in a pleasing way. Were the input signal monophonic, a dynamic stereo field would be created. The stereo field occurs because a time-delay difference between two output channels carrying coherent signals elicits a localization cue. This is known as the Haas effect. While musically interesting, Haas is a persistent source of irritation for recording engineers attempting to place a musical instrument accurately into a stereo mix. For this reason it is prudent to place a stereo field control (or a panning circuit) at the output of any chorus algorithm. It is also useful to have a user switch for disabling the quadrature modulation, that is, for in-phase modulation. (Antiphase is also an option.)

If the flange effect utilizes the same two circuits for stereo processing as did the stereo chorus, then the flanger would normally incorporate delay modulation of the same phase in each channel. Otherwise the comb filtering is diminished due to acoustic mixing in air at the loudspeaker output. The strong flange zone occupies approximately the first 1 ms of the delay line. The viable regions of delay excursion for the chorus and flange effects overlap, however, and can be determined easily by ear. Table 7 gives the approximate delay range and the nominal setting of tap center for the various effects achievable by the circuit of Fig. 36. From the range end one can determine a suitable estimate of the delay-line size N.

It is important to keep the chorus circuit free of nonlinearity for those many guitarists who want the chorus effect as clean and clear as possible. For them, simplicity of the chorus design is a virtue as they use this effect almost all the time. Hence all-pass interpolation[87] for delay modulation becomes critical to the transparency of any chorus. Recall that linear interpolation is a time-varying low-pass filtering process. Indeed, a multivoice (more than two) chorus design using linear interpolation subjects the signal to significantly audible amounts of low-pass filtering attributable to the interpolation. We term the chorus *white* when both negative feedback and all-pass interpolation are used to minimize the spectral aberration that would occur in the absence of these two signal-processing techniques.

Flangers, on the other hand, can benefit from a mild memoryless nonlinearity introduced into the input signal path which resides in front of the entire effect, so that the flange-induced troughs see a richer signal source. But all-pass interpolation is also critical to successful flanger design so as to ensure that the dynamically delayed signal remains unfiltered. The low-pass filtering of the delayed signal, introduced as an artifact of linear interpolation, will reduce the depth of the high-frequency troughs. This is undesirable for a good flanger.

6.2 Appendix 4: Multirate Audio Processes

The purpose of this appendix is to make a rigorous connection between the formal DSP approach to interpolation and decimation, and the time-domain formulation by fractional sample delay as presented in the interpolation sections. The reading of this material is optional and suggested only for those who are already familiar with the viewpoint of Vaidyanathan or Crochiere [29],

[83] NOMINAL_DELAY is usually made to track the depth of vibrato (CHORUS_WIDTH; Fig. 23) in the best vibrato algorithms, that is, the smaller the better.

[84] Previously known as double tracking, a singer would attempt to record the same performance onto a second track while listening to the first. When machines became available to emulate this effect in real time, thereby saving a track, it became known as doubling. The delay modulation is best randomized for the doubling effect

[85] Fixed at about 80 ms, we discovered that you get the Elvis Presley echo effect. George Martin.

[86] Section 7 on sinusoidal oscillators discloses highly efficient quadrature designs.

[87] As discussed in Section 5.

[34], [14, ch. 3.6] and who are comfortable with the z transform.

6.2.1 Upsampling and Downsampling

Fig. 37 shows a real time-domain signal $x[n]$ having the given fictitious spectrum $X(e^{j\omega})$ composed only of a real part. We make this simplifying assumption to improve the clarity of Figs. 38 and 39.

Both upsampling and downsampling are linear time-varying operators. Upsampling [Fig. 38(a)] inserts $L - 1$ zeros between every pair of original samples; no information is lost. Downsampling [Fig. 39(a), $l = 0$] discards $M - 1$ out of every M original samples. In both cases, the original sample presented at time $n = 0$ is passed. L, M, and l are integers; L and M are strictly positive.

Fig. 38 shows the result of subjecting the signal in Fig. 37 to the process of upsampling, whereas Fig. 39 shows the corresponding result due to generalized ($l \neq 0$) downsampling. Notice in both cases that the sample grid is maintained. Thus the associated spectra basically contract and expand, respectively. That is the essence of the method of Vaidyanathan [29].

We now derive the upsampling equation, which has the upsampling factor, L

$$\begin{aligned} Y_I(z) &= \sum_{n=-\infty}^{\infty} y_I[n] z^{-n} \\ &= \sum_{n=-\infty}^{\infty} \left\{ \sum_{p=-\infty}^{\infty} x[p]\delta[n - pL] \right\} z^{-n} \\ &= \sum_{p} \left\{ \sum_{n} x[p]\delta[n - pL] z^{-n} \right\} \\ &= \sum_{p} x[p] z^{-pL} \\ &= X(z^L) . \end{aligned} \tag{42}$$

Eq. (42) is the result of upsampling, expressed in terms of the z transform of the original signal $X(z)$.

We call the downsampler in Fig. 39(a) *generalized* because of the preceding advance operator [29, ch. 4.3, p. 122]. The integer advance l finds use when either positive or negative.

For the example given ($l = 1, M = 2$) we have chosen to ignore the (complex) linear-phase term $e^{j\omega/2}$ in the plot of Fig. 39(c) to make more clear the overlaying of the expanded spectrum. The spectral overlay (more apropos, replication) is demanded by the generalized downsampling equation [Eq. (44)] copied in Fig. 39(a). The linear-phase term corresponds to the half-sample advance in the result, as can be seen by comparing Figs. 39(b) and 37.

The frequency-domain downsampling equation [Eq. (44)] in Fig. 39(a) is a true z transform that is derived from the time-domain expression for downsampling in Fig. 39(b), once we make the substitution

$$\sum_{p=-\infty}^{\infty} \delta[nM - p] = \frac{1}{M} \sum_{k=0}^{M-1} W_M^{-kp} , \qquad -\infty < n < \infty . \tag{43}$$

Here $\delta[n]$ is the Kronecker delta function. The identity Eq. (43) is the bridge between the two domains.

We now derive the downsampling equation Eq. (44), which has the downsampling factor M, the equation being generalized by l an arbitrary integer advance as in Fig. 39(a). Taking the z transform,

$$Y_D(z) = \sum_{n=-\infty}^{\infty} y_D[n] z^{-n} = \sum_{n=-\infty}^{\infty} x[nM + l] z^{-n} .$$

First we substitute the equivalent form of the signal $x[nM + l]$, whose time index remains n. Then we apply Eq. (43),

$$\begin{aligned} Y_D(z) &= \sum_{n=-\infty}^{\infty} \left\{ \sum_{p=-\infty}^{\infty} x[p + l]\delta[nM - p] \right\} z^{-n} \\ &= \sum_{p/M=-\infty}^{\infty} \left\{ x[p + l] \frac{1}{M} \sum_{k=0}^{M-1} W_M^{-kp} \right\} z^{-p/M} \\ &= \frac{1}{M} \sum_{k} \sum_{p=-\infty}^{\infty} x[p + l] z^{-p/M} W_M^{-kp} . \end{aligned}$$

Since the sum over k only has value when $p = nM$ by Eq. (43), there is no complex root of z taken. Hence that exponentiation is unambiguous. Now we let $p + l \rightarrow r$,

$$\begin{aligned} Y_D(z) &= \frac{1}{M} \sum_{k} \sum_{r=-\infty}^{\infty} x[r] z^{-(r-l)/M} W_M^{-k(r-l)} \\ &= \frac{1}{M} \sum_{k=0}^{M-1} (z^{1/M} W_M^k)^l X(z^{1/M} W_M^k) \end{aligned} \tag{44}$$

Eq. (44) is the desired result, which appears in Fig. 39(a), expressed in terms of the z transform of the original signal $X(z)$.

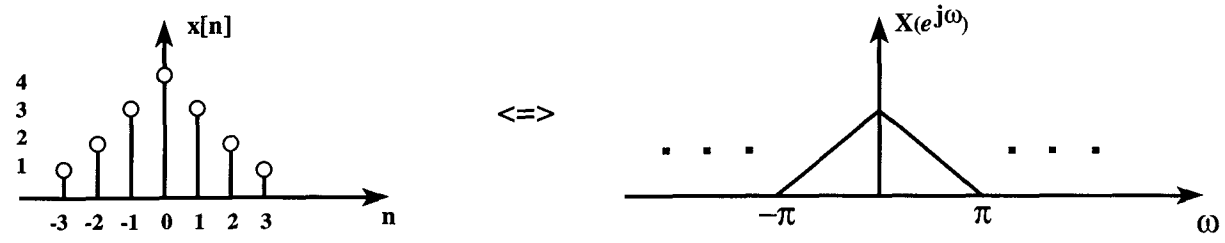


Fig. 37. Time-limited input signal and its fictitious real spectrum.

6.2.2 Conversion: Vaidyanathan to Crochiere

The Vaidyanathan method of analysis is simpler because it ignores changes in sample rate, whereas the method of Crochiere [34] does not, that is, the Vaidyanathan method treats any sampled signal as an indexed sequence [29, ch. 4.1.1, p. 111]. But note that in terms of the insertion or deletion of samples, the action of the upsampler block [Fig. 38(a) and (b)] or the downsampler block [Fig. 39(a) and (b)] is respectively the same using either method.

While the two methods of analysis are equally valuable, it is prudent to have a means of converting between the two. Given the Vaidyanathan analysis, one simply substitutes every occurrence of z with z' (Crochiere's notation) which is defined equal to $z^{M/L}$. One would also relabel any frequency-domain graphs by substituting ω' (which equals $\omega M/L = \Omega T\, M/L$) for ω.

The conversion becomes exceedingly simple when either L or M equals 1. For example, we wish to convert Fig. 38 ($M = 1, L = 3$) to the Crochiere-style upsampler analysis. Then we must relabel the abscissa, using ω' in place of ω, and we relabel the ordinate as $X(e^{j\omega' L}) = X(e^{j\omega})$ in place of $X(e^{j\omega L})$. This result is correct because

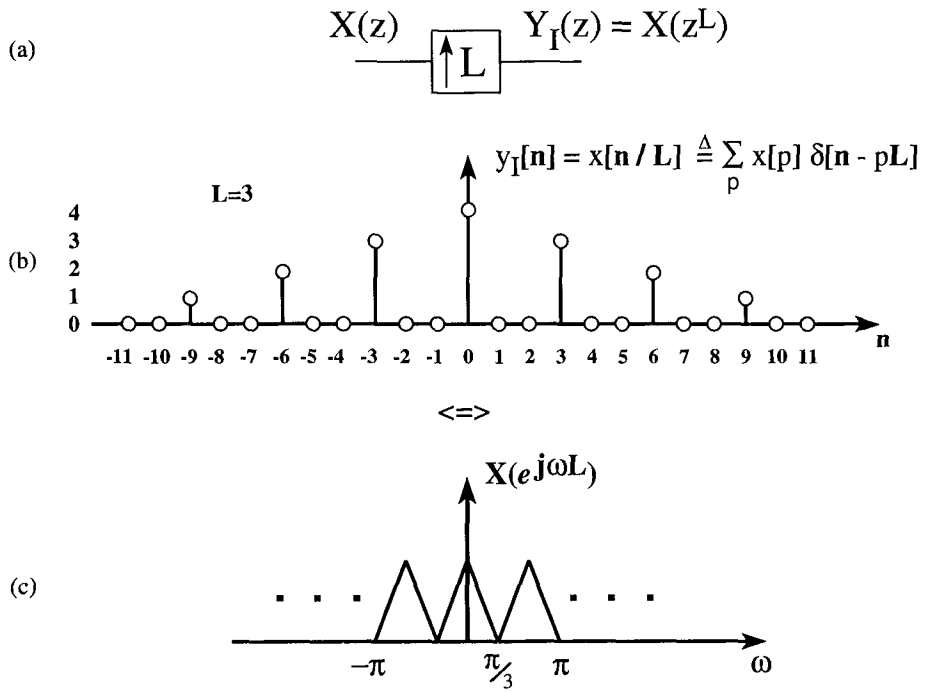


Fig. 38. Upsampler (Vaidyanathan) [29].

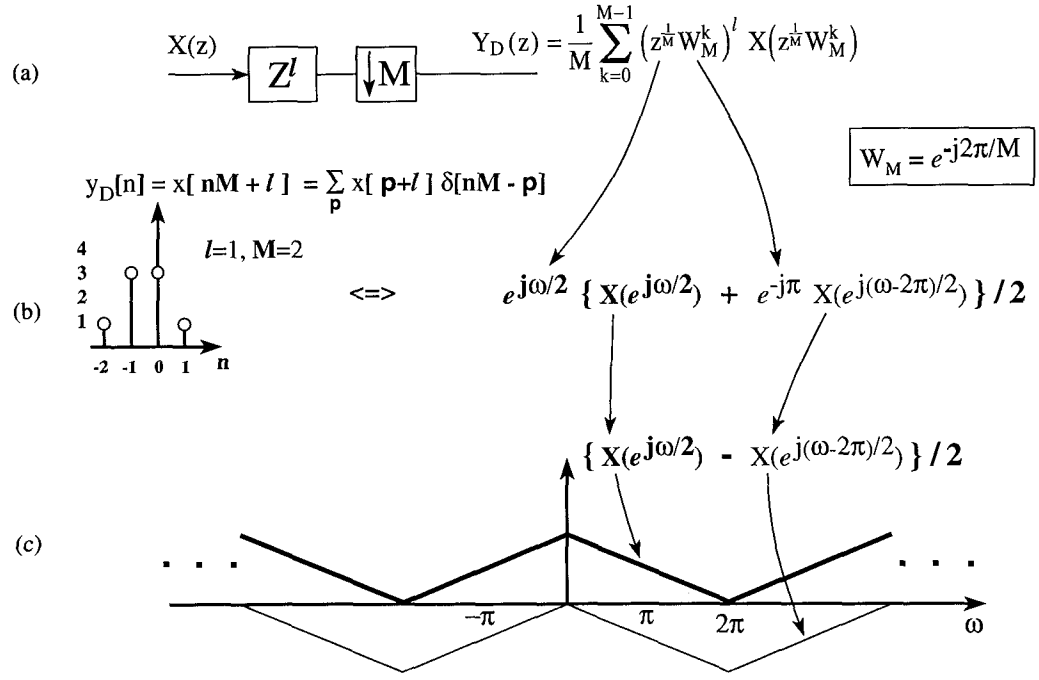


Fig. 39. Generalized downsampler (Vaidyanathan) [29].

Crochiere always maintains the absolute time period between the original samples. This is true for both upsampling and downsampling.

6.2.3 Interpolation and Decimation

Upsampling and downsampling, each combined with the appropriate digital filter, describe the processes of interpolation and decimation, respectively. It is possible to combine the two processes by sharing the same digital filter, as illustrated in Fig. 40. The two equivalent figures (Figs. 40 and 41) tie together the formal analysis in terms of the sample rate conversion ratio L/M and our preferred interpretation of the interpolation and decimation processes in terms of the required fractional sample delay [34, pp. 40, 81]. The role of the upsampler is played by the commutator in Fig. 41 [29, pp. 131, 124]. Theoretically, each polyphase filter $E_l(z)$ in Fig. 41 offers a different but fixed fractional sample delay to an incoming signal, that is, each is ideally all pass.

We often refer to the complete process depicted in Fig. 40 or in Fig. 41 simply as interpolation. That is an abuse of terminology, strictly speaking, as both the process of interpolation and that of decimation are carried out in those figures, having $H(z)$ simultaneously serve both purposes.

Let us eliminate a possible point of confusion. When we introduced the polyphase filter in Sections 4 and 5, it was presented as a time-varying circuit having two inputs (see Figs. 24, 29, and 32). We have made an analytical connection, or perhaps a leap, to the classical network of Fig. 41. When we insert one of those polyphase filter topologies into the many branches of the commutator circuit in Fig. 41, they curiously become time invariant, and the second input goes away by connecting the dangling unit delay. The justification for these modifications is simply this. The original polyphase filter circuits were time varying and dual input because we were trying to find a simpler way to implement the entire complicated commutator network that is Fig. 41.

Nonetheless, we can justifiably say that the most efficient technique for performing interpolation is through polyphase filtering. Fig. 41 is then the classical circuital definition of polyphase filtering because it shows how time-invariant single-input polyphase filters comprise the general linear time-varying frequency-domain model of interpolation represented in Fig. 40.

6.2.4 Polyphase Filter[88]

In this section we first determine the polyphase filters, then from them we find what Vaidyanathan and Crochiere refer to as the *prototype* interpolation filter $H(z)$. This approach is backward with respect to the traditional design method[89] because that is how the time-domain formulation by fractional sample delay proceeds. In Sections 4 and 5, only the polyphase filters are considered. We are forced to accept whatever prototype is tacit. In Fig. 40, however, $H(z)$ prescribes the formal prototype interpolation filter as an antialiasing filter when $L < M$ or as a replication filter when $L > M$.[90]

Each time-invariant polyphase filter, represented by $E_l(z)$ in Fig. 41, presents a different fractional sample delay to a signal. More accurately, for the idealized formulation of interpolation by a rational factor, each polyphase filter is exactly all pass, noncausal, and linear phase,

$$E_l(e^{j\omega}) = e^{j(\omega - 2\pi m)l/L}, \qquad 2\pi m - \pi < \omega < 2\pi m + \pi$$

$$= e^{j2\arctan[\sin\omega/(1+\cos\omega)]l/L}, \qquad 0 \leq l < L \qquad (45)$$

where l is the polyphase filter number, L is the upsampling factor, and m is an integer. The desired phase of

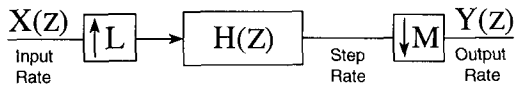


Fig. 40. General model of interpolation by a rational factor.

[88] Our definition is from [34, ch. 3.3.2, p. 80].

[89] The required low-pass specifications in the traditional method of prototype design are stated at the end of this appendix.

[90] Note that no filtering is necessary when $L = M$.

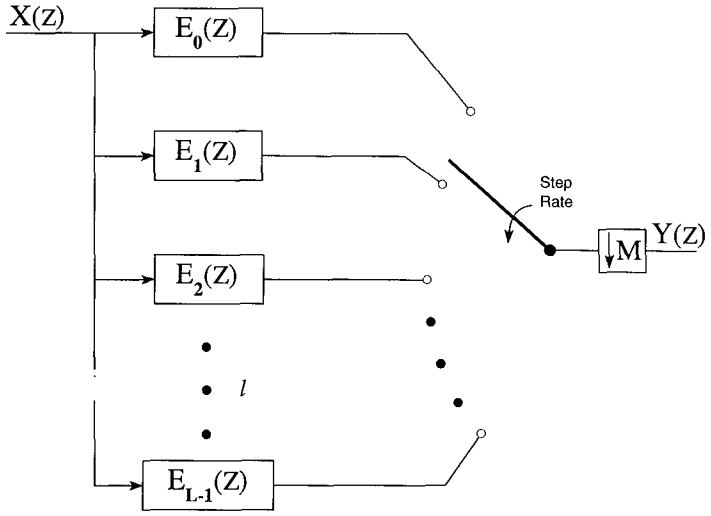


Fig. 41. Commutator model of classical polyphase interpolation showing numerous polyphase filters.

all the ideal polyphase filters [Eq. (45)] is linear [29, ch. 4, pp. 168, 109, 124], [34, ch. 4.2.2], [35], [37]. Eq. (45) defines linear phase in a discrete-time system. Hence the lth ideal polyphase filter represents a time *advance* of l/L fractional samples that is independent of frequency. Refer to Fig. 42, where the nonwrapped linear phase is represented.[91]

This idealization presumes that the prototype interpolation filter is perfectly band-limited to π/L (on one side). To construct the prototype from these ideal polyphase filters [Eq. (45)] we use the important relation,[92]

$$H(z) = \sum_{l=0}^{L-1} z^{-l} E_l(z^L) \,. \tag{46}$$

Now for every modulo $2\pi/L$ of ω, m increments. So we have

$$H(e^{j\omega}) = \sum_{l=0}^{L-1} e^{-j\omega l} e^{j(\omega L - 2\pi m)l/L}$$

$$= \sum_{l=0}^{L-1} e^{-j2\pi ml/L}, \qquad 2\pi m - \pi < \omega L < 2\pi m + \pi \,.$$

From Eq. (43), this only has value, in the baseband, for $m = 0$. Hence,

$$H(e^{j\omega}) = \begin{cases} L \,, & |\omega| < \dfrac{\pi}{L} \\ 0 \,, & \dfrac{\pi}{L} < |\omega| < \pi \,. \end{cases} \tag{47}$$

This zero-phase prototype $H(e^{j\omega})$ is known as an ideal Lth-band [Nyquist(L)] filter.[93] Its bandwidth, as indicated in Eq. (47), will admit aliasing when $L < M$. This may seem antithetical, but we must remember that the idealized formulation of interpolation simply finds the requested point between two original samples of the analog signal, as in Eq. (39). It does not purposely alter the bandwidth of the original signal before finding that new point. Analogously, if we were to sample an analog signal at too slow a rate, its spectrum would alias.

The zero-phase prototype interpolation filter [Eq. (47)] has a noncausal impulse response in the form of the classical analog sinc( ) as $L \to \infty$. Hence this idealized formulation of interpolation employing noncausal linear-phase all-pass polyphase filters corresponds to a resampling of the classical reconstruction [14, ch. 3.3],

$$h[n] = \frac{\sin(\pi n/L)}{\pi n/L} \,.$$

Fig. 43 shows this $h[n]$ corresponding to Eq. (47). The polyphase filters $E_l(e^{j\omega})$ [Eq. (45)] can be visualized in the time domain via Eq. (53). Each is a subsampled likeness of $h[n]$. These $E_l(e^{j\omega})$, then, are the ideal polyphase filters of classical polyphase all-pass interpolation, which is implemented as illustrated in Fig. 41. We will devise a warped and a nonwarped all-pass polyphase filter realization, which are somewhat close to the ideal.

6.2.5 Theoretical Values of L and M

When the interpolation process is discussed as in Sections 4 and 5 in terms of the required fractional sample delay, the upsampling factor L is implicit in the sheer resolution of the filter coefficients (frac and 1 − frac), that is, the delay resolution is determined by the coefficient resolution (quantization) and vice versa. Hence the implicit value of L is 2^{23} because we require no less than a 24-bit processor for audio [1], [12], [49].

When we ponder the interrelationship of the respective frequency-domain and time-domain analyses of Figs. 40 and 41, it becomes clear that L determines the upsampling factor as well as the number of polyphase filters,

[91] Fig. 42 depicts, in fact, the classical discrete-time definition of linear phase, which is given within Eq. (45). The phase periodicity is not made explicit in many textbooks on DSP.

[92] It may seem that we have pulled Eq. (46) out of a hat. It is because we will accept Eq. (46) as the definition of all polyphase filters. Rather, Eq. (46) states how the polyphase filters are expected to be combined to form the prototype interpolation filter [29, ch. 4.3].

[93] If L were 2 it would be known as an ideal half-band filter [29].

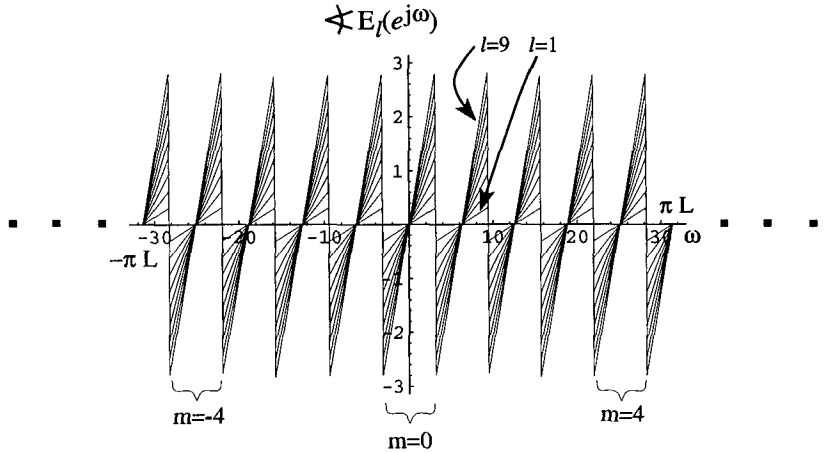


Fig. 42. Phase response of ideal polyphase filters $E_l(e^{j\omega})$; $L = 10$.

hence the delay resolution. This fact is self-evident in the ideal case [Eqs. (47) and (45)]. Recalling Eq. (41) we may now make the connection between the ideal and the realizable polyphase filters,

$$\tau = \text{frac} = \frac{l}{L} \tag{48}$$

where τ is the desired fractional sample delay, frac is the elemental polyphase filter coefficient, l is the commutator branch number, and L is the upsampling factor.

For the particular applications of delay modulation that were discussed at length, the pitch change ratio M/L is neither large nor small. Hence M is about the same order of magnitude as L. When M and L have common factors, it is valid to reduce the fraction M/L until M and L are mutually prime. The interpolation that results is identical.

Fig. 41 is primarily of theoretical interest then, since 2^{23} fixed coefficient sets are not easily stored. What we found in Sections 4 and 5 were easy ways to compute the required coefficients in real time, and how to apply them to one dual-input time-varying filter (see Figs. 24, 29, and 32). But then we discovered that transient distortion was generated when the filter is recursive (see Figs. 29 and 32) because we had no way to keep track of 2^{23} states (the required number of recursive memory elements). We will show how to draw that lone dual-input recursive filter circuit as an equivalent network resembling that of Fig. 41 (see Fig. 50).

pler. Only $L/\text{GCD}(L, M)$ of them will be seen. Of those that do appear, each will be accessed with a frequency of

$$\frac{F_s L}{\text{LCM}(L, M)} \leq F_s$$

where LCM is the least common multiple and GCD is the greatest common divisor. Reversing the brush direction would simply change one sign of the z exponent in Eq. (49), namely, to z^{l-l_0}.

The individual $E_l(z)$ in Fig. 41 are noncausal,[95] linear time invariant, and correspond to one of the causal circuits shown in Fig. 25, and in Fig. 29 or Fig. 32 (both having the delay element connected) for the cases of linear[96] and classical polyphase all-pass interpolation, respectively. Recalling the nomenclature we previously devised for those circuits in Eq. (39), we fix the coefficients of the lth polyphase filter by making the identification in Eq. (48).[97]

In the case of linear interpolation, $E_l(z)$ is a first-order (two-tap) noncausal FIR filter by design,

$$\left.\begin{aligned} P_l(z) &= (1 - l/L) + (l/L)z^{-1} \\ E_l(z) &= zP_{L-l}(z) \end{aligned}\right\} \quad \text{linear interpolation}$$

while in the case of classical polyphase all-pass interpolation we realize a circuit close to the ideal polyphase filter [Eq. (45)] using a first-order all-pass filter made noncausal,

$$\left.\begin{aligned} P_l(z) &= \frac{(1 - l/L) + z^{-1}}{1 + (1 - l/L)z^{-1}} \\ E_l(z) &= zP_{L-l}(z) \end{aligned}\right\} \quad \text{classical polyphase all-pass interpolation}$$

$$\left.\begin{aligned} P_l(z) &= \frac{(1 - l/L)/(1 + l/L) + z^{-1}}{1 + [(1 - l/L)/(1 + l/L)]z^{-1}} \\ E_l(z) &= zP_{L-l}(z) \end{aligned}\right\} \quad \begin{aligned} &\text{classical polyphase all-pass interpolation} \\ &\text{with coefficient warping [Eq. (40)].} \end{aligned}$$

These three $P_l(z)$ are the causal transfer functions of the circuits in Figs. 25, 29, and 32 (the latter two having

6.2.6 Commutator Circuit

The transfer function of the linear time-varying commutator filter circuit in Fig. 41 (not including the downsampler) can be described using a generalization of Eq. (46),

$$H(z) = \sum_{l=0}^{L-1} z^{-(l-l_0)} E_l(z^L) \tag{49}$$

where l_0 is the starting phase of the commutator, that is, the initial position of the brush when the input sample $x[n]$ arrives. The action of the downsampler[94] is synchronized to l_0 such that the very first sample found at position l_0 is passed to the output. Depending on the particular values of L and M, not necessarily all of the polyphase filters $E_l(z)$ will be seen at the output of the downsam-

[94] Taking the view of Crochiere and Rabiner, for the moment, the downsampler would operate at the step rate LF_s [34].

[95] Considering the noncausal polyphase filter $E_l(z)$ simplifies the exposition and keeps us in agreement with Vaidyanathan's development of the subject. We will let $P_l(z)$ denote the causal polyphase filter.

[96] Because the many polyphase filters of linear interpolation require no long-term memory, we were able to cheat the classical implementation by using the equivalent time-varying circuit shown in Fig. 24 instead of the aggregate network in Fig. 41. Hence we perfectly implement classical polyphase linear interpolation with ease. (We drop the term "classical polyphase" in the discussion.)

[97] The number of polyphase filters L is determined by the numerical resolution of the registers holding the polyphase filter coefficients; that is, there are as many polyphase filters as there are different possible quantized values of coefficients. Note that because of the method of implementation (refer to Fig. 23, for example) in all the polyphase applications discussed, there is no division required in the capture of the elemental coefficient l/L.

the delay element connected) [44, p. 178], [39, p. 50]. All three transfers are time invariant when the corresponding circuits are inserted into the lth branch of the commutator circuit in Fig. 41.

We reverse the order of the index in $P_l(z)$ via the substitution $l \rightarrow L - l$ because the polyphase filters $E_l(z)$ in Fig. 41 are ordered by increasing time advance, whereas the $P_l(z)$ are ordered by increasing delay.

The ideal $E_l(z)$ are noncausal [Eq. (45)]. This warrants multiplication of the various first-order causal $P_{L-l}(z)$ by z so that the impulse response of each corresponding prototype filter Eq. (46) is time aligned with the others (see Figs. 43–46). But this analytical z factor is unnecessary when realizing an implementation.

6.2.7 Tacit Prototypes H(z)

For the prototype interpolation filter of linear interpolation, the length of its triangular impulse response is $2L$, as illustrated in Fig. 44. That finite length, by design, spans only two of the original input sample periods [14, ch. 3.6.2, p. 109], whereas in the case of classical polyphase all-pass interpolation, the impulse response is of infinite length, as implied in Figs. 45, 46, and 43. All the impulse responses in the figures were determined using the signal processing package of Evans et al. [25] for *Mathematica* to invert Eq. (46) directly.

All the impulse responses we show have every Lth sample equal to zero (except the central sample), thus identifying Lth-band [Nyquist(L)] prototype filters [29, ch. 4.6], [35]. Hence the original input samples $x[n]$ appear unscathed[98] at the input to the downsampler in Fig. 41, regardless of l_0.

The initial linear slope of the impulse response for the all-pass interpolation prototype in Fig. 45 is identical to that for the linear interpolation prototype in Fig. 44. This can be explained best by observing Fig. 41 for the case $M = 1$, $L = 10$. The commutator brush visits all the polyphase filters within the same time period of one input sample. If the input signal $x[n] = \delta[n]$ and $l_0 = 0$, then what we shall see at the brush is the impulse processed by each of the all-pass polyphase filters successively. But on this first sweep of the brush, none of the all-pass filters has anything stored in its memory,[99] so the filters are effectively scalar multipliers increasing linearly with l.

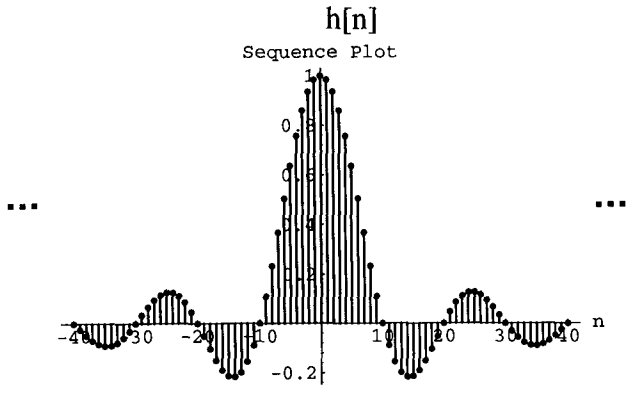


Fig. 43. Ideal prototype interpolation filter impulse response; $L = 10$.

Another salient characteristic of Lth-band filters is that the frequency-shifted filter sums to a constant level in the frequency domain,

$$\sum_{k=0}^{L-1} H(zW_L^k) = LE_0(z^L) = L \tag{50}$$

which is true since we have $E_0(z) = 1$ for all our polyphase filters.[100]

Next we show these Lth-band prototype filters in the frequency domain via Eq. (46). Figs. 47, 48, and 49 give the prototype filter magnitude responses for linear, all-pass, and warped all-pass interpolation, respectively, for the case $L = 10$. For $2L$-length linear interpolation we expect a positive real frequency response of the form

$$H(e^{j\omega}) = \frac{\sin^2(\omega L/2)}{L \sin^2(\omega/2)}. \tag{51}$$

The expressions for the all-pass interpolation prototypes are not so simple, neither are their nonzero phases [35].

As we have already seen, there is a mathematical description of a prototype interpolation filter in terms of its polyphase filters,

$$H(z) = \sum_{l=0}^{L-1} z^{-l} E_l(z^L). \tag{46}$$

The converse is also true,

$$E_l(z^L) = \frac{1}{L} \sum_{k=0}^{L-1} (zW_L^k)^l H(zW_L^k) \tag{52}$$

$$= \sum_{n=-\infty}^{\infty} h[nL + l]\, z^{-nL}. \tag{53}$$

Without significant loss of the generality of Eq. (49), Eq. (46) describes well the filtering action of the commutator circuit presuming $l_0 = 0$. Eq. (52), a generalization of Eq. (50), is an application of the generalized downsampling equation (44) to $H(z)$ for advance l. Eq. (52) applies directly to the polyphase decomposition of either IIR or FIR prototypes. The time-domain impulse response $h[n]$ in Eq. (53) of course corresponds to $H(z)$ while $h[nL + l] \leftrightarrow E_l(z)$.

[98] If we set $M = 1$ just for convenience, then we can use the y nomenclature to make this point. In that case, $y[nL + (L - l_0) \bmod L] = x[n]$. When $l_0 = 0$, we have the simpler result $y[nL] = x[n]$.

[99] Refer to Fig. 29, having the delay element connected.

[100] A proof of Eq. (50) is suggested by expressing $H(z)$ as in Eq. (46). We will generalize Eq. (50) in Eq. (52).

6.2.8 Complete Commutator Network Transfer Function

Having established the definitions of Eqs. (52) and (46), we can write the equation for the commutator model of Fig. 41 in the manner of Vaidyanathan,

$$Y(z) = [X(z^L)H(z)] \downarrow_M$$

$$= \frac{1}{M} \sum_{m=0}^{M-1} X(z^{L/M} W_M^{Lm}) \sum_{l=0}^{L-1} (z^{1/M} W_M^m)^{-l} E_l(z^{L/M} W_M^{Lm}) . \quad (54)$$

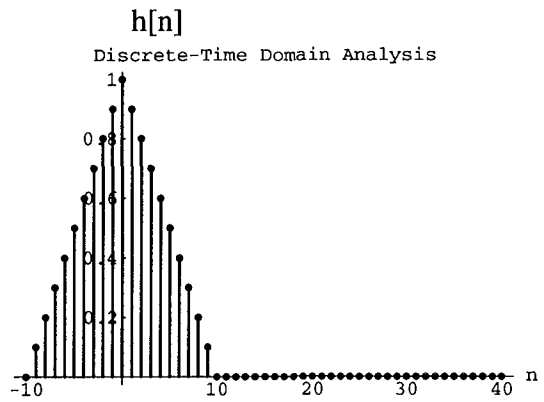


Fig. 44. Impulse response of linear interpolation prototype filter; $L = 10$.

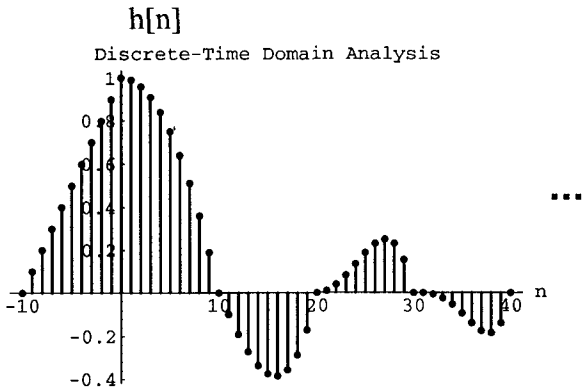


Fig. 45. Classical polyphase all-pass interpolation prototype filter impulse response; $L = 10$.

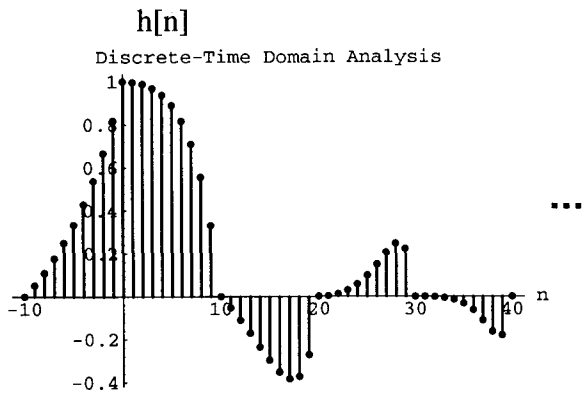


Fig. 46. Classical polyphase warped all-pass interpolation prototype filter impulse response; $L = 10$.

This formidable equation reduces nicely for special cases of interest,

$$Y(z) = X(z)E_0(z) , \qquad M = L$$

$$= X(z)$$

recalling that $E_0(z) = 1$.

$$Y(z) = X(z^2)[E_0(z^2) + z^{-1}E_{L/2}(z^2)] , \qquad M = L/2$$

$$= X(z^2)[1 + z^{-1}E_{L/2}(z^2)] .$$

In this case the two polyphase filters are attempting to form the half-band filter required of an effective interpolation by the factor $2 = L/M$ (pretending that $M = 1$).

The resulting expressions are not as simple when

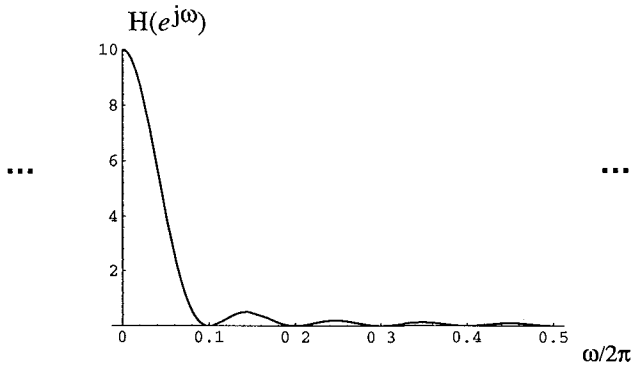


Fig. 47. Linear interpolation prototype filter frequency response; $L = 10$.

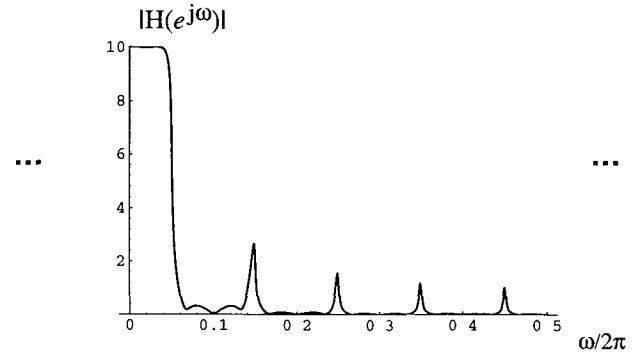


Fig. 48. Classical polyphase all-pass interpolation prototype filter magnitude response; $L = 10$.

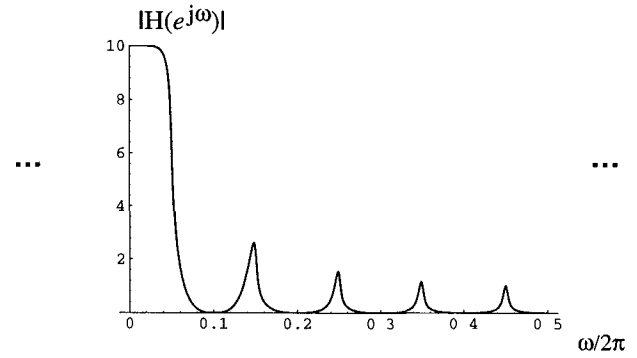


Fig. 49. Classical polyphase warped all-pass interpolation prototype filter magnitude response; $L = 10$.

L/M is irreducible. The fascinating aspect of Eq. (54), however, is that it gives us a closed-form representation of the linear time-varying network, namely, Fig. 41. The reader is encouraged to explore that further[101] and thereby get a better feel for the relationship of Eq. (54) to the commutator model in Fig. 41.

6.2.9 Engineering Approximation to Classical Polyphase All-Pass Interpolation

The polyphase filter circuits as first presented in the sections on interpolation (Sections 4 and 5) were time varying and dual input because we were trying to cheat and to find a simple way to implement the complicated but desirable commutator network that is Fig. 41. We were only partially successful because in the case of all-pass interpolation we could only implement Fig. 50, which is a circuital approximation to Fig. 41. But for linear interpolation we were able to implement Fig. 41 exactly; that was our inspiration.

The many branches of the commutator network in Fig. 41 contain time-invariant single-input polyphase filters. Rather than implement that intensive classical network for the case of all-pass interpolation, our "innovation" (as proposed and demonstrated in Section 5) is to implement instead the simpler formal network in Fig. 50. Fig. 50 is a circuital approximation to Fig. 41 via reduction in the amount of recursive memory required. By sharing the recursive memory of all the all-pass polyphase filters in Fig. 41, we eliminate the recursive memory from the left-hand side of the commutator as we move it to the right-hand side as shown in Fig. 50. Consequently we are able to exactly implement the approximating network in Fig. 50 using only one equivalent time-varying dual-input all-pass filter, as shown in Fig. 29 or Fig. 32.

The recursive circuit at the output in Fig. 50 is time varying. Its coefficient tracks the commutator brush. The recursive elements have been eliminated from the polyphase filters of Fig. 50.[102] Each $E_l(z)$ is single input and nonrecursive, having no long-term memory. Now their time-invariant transfer functions are

$$\left.\begin{aligned} P_l(z) &= (1 - l/L) + z^{-1} \\ E_l(z) &= zP_{L-l}(z) \end{aligned}\right\} \quad \text{all-pass interpolation approximation .}$$

Fig. 50 is the equivalent of Fig. 29 *as is*, that is, as Fig. 29 is drawn—time varying and dual input. Everything to the left of the commutator in Fig. 50 is time invariant. Note that because of insufficient memory, the lone time-varying recursive circuit on the right-hand side of the commutator can only approximate the L different time-invariant recursive circuits that are supposed to be associated with the $E_l(z)$ on the left-hand side, as in Fig. 41. This is the nature of the circuital approximation.

As we saw in Section 5, this circuital approximation in Fig. 50 to the classical polyphase all-pass interpolator of Fig. 41 is only good, in terms of ***THD + N***, for microtonal changes in pitch, that is, for M/L near 1.

One further refinement is our warping of the all-pass polyphase filter coefficients, computed in real time[103] and used in the circuital approximation (Fig. 50) to the classical polyphase network. The polyphase filter coefficients change slightly,

$$\left.\begin{aligned} P_l(z) &= \frac{1 - l/L}{1 + l/L} + z^{-1} \\ E_l(z) &= zP_{L-l}(z) \end{aligned}\right\} \quad \text{all-pass interpolation approximation with coefficient warping [Eq. (40)] .}$$

With warping, Fig. 50 becomes equivalent to Fig. 32 as is. Using this refinement, we measured an improvement of 26 dB in ***THD + N*** for microtonal pitch changes. (This has already been discussed in Section 5.)

6.2.10 Distortion, Transients, and Large Pitch Change

Distortion is inherent in any interpolation process. For large pitch change with little distortion, one's choice (in our context) is to revert to linear interpolation at a very high sample rate or to implement the formal network in Fig. 41 by brute force for classical polyphase warped all-pass interpolation. We have simulated Fig. 41 in the C programming language, where we use the warped all-pass polyphase filter of Fig. 32 having the delay element connected and time-invariant 16-bit coefficients. To achieve excellent results over a large range of M/L for classical polyphase all-pass interpolation, we find that coefficient warping [Eq. (40)] is necessary and that about $L = 2^8$ recursive states must be stored.

Since there are no time-varying coefficients in the classical polyphase network of Fig. 41, associated transient phenomena are nonexistent components of the signal distortion characteristics. Transients that do arise there are due to the zero-state response (ZSR) of recursive polyphase filters. The ZSR transients will be short for polyphase filters having small feedback coefficients, and longer for larger coefficients.

When making ***THD + N*** estimates for classical poly-

[101] Keep in mind the identity Eq. (43).

[102] The idea of moving the recursive part of the polyphase network outside of the commutator is not new. Crochiere and Rabiner [34, ch. 3.4] proposed separating out such a polyphase filter denominator, but their denominator was time invariant and common to all polyphase filters.

[103] This step is optional; review Eq. (40).

phase warped all-pass interpolation, we ignore the first several samples output so as to avoid ZSR transients. The C program simulation averages about −89 dB of *THD+N* under the same signal conditions as those of Fig. 35(b).

6.2.11 Polyphase Perspectives

The sample rate conversion ratio *L/M* in Fig. 51(a) corresponds to the pitch change ratio inverse. We presume that the pitch change process in Fig. 51(b) is performed on a stored sample record. Such would be the case in contemporary sampler-type synthesizers. Samples effectively appear at the input to the pitch change effect at a rate different from which they were recorded. The original duration of the sample record is altered by the ratio *L/M*.

The Vaidyanathan method of analysis is performed independent of the sample rate, as in Fig. 51(c). Rates of computation are determined as a post-operative step if necessary. (Refer to Section 6.2.2 in this appendix.) Since the sample record is considered to be an indexed sequence, any output rate can be ascribed.

6.2.12 Application Schema

Time compansion (compression and expansion) was accomplished before the days of DSP by physically splicing fragments of magnetic recording tape to alter the run time without the concomitant change in perceived pitch [50]. Contemporary compansion machines control the playback speed of the recording medium while performing the manually tedious algorithm deftly in real time. This process is shown in Fig. 52(a). The splicer algorithm regulates equestrian jumps by ρ pointer to compensate for the disparity in sample rate between the input ω and the output ρ of a delay line.[104]

The process called pitch shift is typically accomplished as shown in Fig. 52(b), also using a splicer. The goal of the pitch shift algorithm is the converse to that of time compansion, that is, to alter the perceived pitch with no change in run time.

The time compansion device may be used to perform pitch shift when the playback speed is left running at normal. This fact has unfortunately obscured the distinction between time compansion and pitch shift devices, which actually employ different means as Fig. 52(a) and (b) illustrates. For this reason, pitch shift effects are often found outside their intended arena, performing the task of a time compansion algorithm—a task for which they are not qualitatively optimized.

In Fig. 52 both the pitch shift and the vibrato [(c)] processes are performed in real time, and, unlike the fixed pitch change algorithm shown in Fig. 51(b), both maintain the macrotemporal features of the original signal via propitious application of a delay line. The splicer is not required for vibrato since the mean sample rate across the delay line is F_s by design, that is, the downsampling factor *M* is sinusoidally time varying.

All three processes shown in Fig. 52 require some

[104] Each jump target is determined by a very high-speed custom autocorrelator seeking periodicity within the delay-line contents. Cross fading is employed by the splicer using two quadrants of a raised cosine. While some time compansion algorithms employ delay-line interpolation, the Lexicon model 2400 stereo audio time compressor/expander (designed in 1986 and still in production [51]) sidestepped the need for interpolation by incorporating a variable-rate A/D conversion system. In that design, the D/A circuitry is fixed rate, the whole instrument operating in real time on prerecorded material.

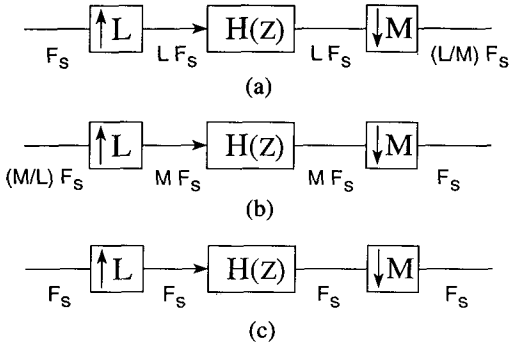


Fig. 51. Contrasting presumed computation rates of various methods of analysis. (a) Sample-rate conversion; Crochiere analysis [34]. (b) Pitch change by fixed amount, *M/L*. (c) Sample-rate conversion; Vaidyanathan analysis [29].

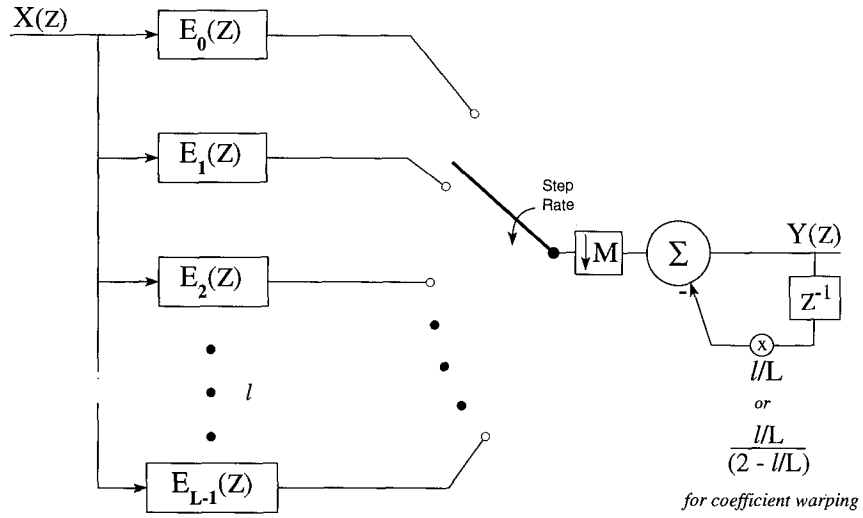


Fig. 50. Commutator model of proposed all-pass interpolator.

significant amount of nominal (average) transport delay through the delay line. To perform well upon polyphonic music, the time compansion and pitch shift algorithms require as much as 60 ms. That much delay is easily perceptible, and a compromise is nearly always necessary. Vibrato, on the other hand, can be performed well using only about 1-ms nominal delay.

When slow microtonal vibrato is mixed with the original signal, we get the chorus effect. When fixed microtonal pitch *shift* is mixed with the original signal, we get the detune effect. The sonic impact of detune is so powerful that it deserves notoriety. The result is often described by musicians as subjectively "fattening" the sound. The chorus and detune effects are sonically quite distinct, the latter algorithm being more difficult to implement properly. The primary distinguishing feature of the two is that the pitch is necessarily undulating in the chorus effect because of the means of implementation. This detune effect occurs naturally and is built into instruments such as the pianoforte, mandolin, and twelve-string guitar. It accounts for one salient character of each instrument's characteristic sound. The reader should also be aware that contemporary sampling music synthesizers[105] often emulate detuning through the use of fixed pitch change, which in many cases is undesirable because the temporal features fall out of alignment in the mix.

6.2.13 Prototype Filter Design

Fig. 53 gives the ideal specifications of the prototype interpolation filter when absolutely no aliasing of the input signal, having one-sided bandwidth σ, is tolerated[106] [29, ch. 4.1.1, p. 109], [34, ch. 2.3.4]. Implicit from the axis labeling is that the desired filter is real in the frequency domain. The filter amplitude L maintains absolute signal level.

Note from Fig. 53 that

$$\frac{\omega_p + \omega_s}{2} = \begin{cases} \frac{\pi}{M}, & \frac{\pi}{M} < \frac{\sigma}{L} \\ \frac{\pi}{L}, & \text{otherwise} . \end{cases}$$

The traditional approach to interpolation would begin with the design of the prototype as specified here, whereas the techniques of interpolation that we considered in Sections 4 and 5 did not allow anywhere near this level of control. We simply accepted what those implementations offer because they are computationally attractive. What they offer is a prototype having a fixed radian cutoff frequency equal to π/L (see Figs. 47–49).

[105] Rather sampler type, which we distinguish from wavetable-type synthesizers.

[106] When $\pi/M < \sigma/L$, this traditional prototype design further band-limits the signal, unlike the techniques we examined in Sections 4 and 5, which admit aliasing instead.

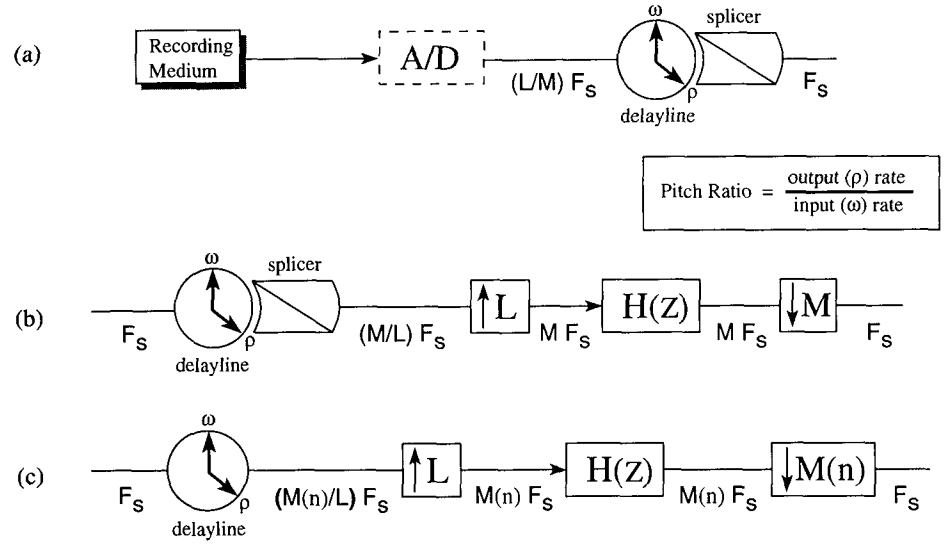


Fig. 52. Applications of discrete-time interpolators. (a) Time compansion. A/D optional (b) Pitch shift. (c) Undulating pitch change; vibrato.

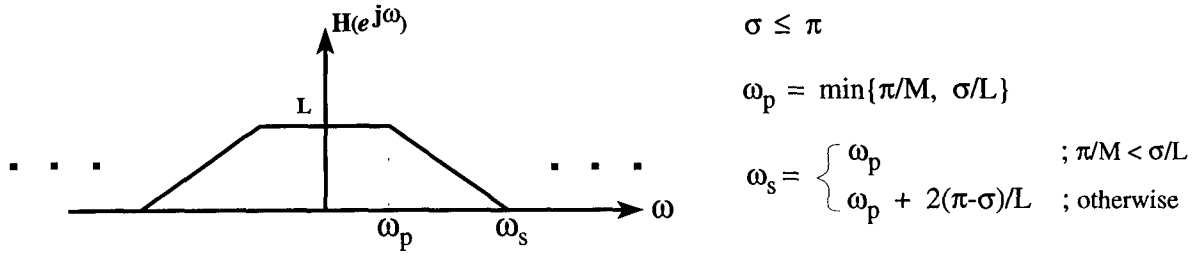


Fig. 53. Prototype interpolation filter specifications for signal bandwidth σ.

REFERENCES

[1] D. C. Andreas, J. Dattorro, and J. W. Mauchly, "Digital Signal Processor for Audio Applications," U.S. patent 5,517,436 (1996 May 14).

[2] W. G. Gardner, "Reverberation Algorithms," in *Applications of Signal Processing to Audio and Acoustics*, M. Kahrs and K. Brandenburg, Eds. (Kluwer Academic, Norwell, MA, 1997).

[3] J. A. Moorer, "About this Reverberation Business," *Computer Music J.*, vol. 3, pp. 13–28 (1979 June), also in *Foundations of Computer Music*, C. Roads and J. Strawn, Eds. (MIT Press, Cambridge, MA, 1988).

[4] D. Griesinger, "Practical Processors and Programs for Digital Reverberation," in *Audio in Digital Times, Proc. Audio Eng. Soc. 7th Int. Conf.* (Toronto, Ont., Canada, 1989 May 14–17), pp. 187–195.

[5] B. A. Blesser and K. O. Bader, "Electric Reverberation Apparatus," U.S. patent 4,181,820 (1980 Jan. 1).

[6] D. Griesinger, "Room Impression, Reverberance, and Warmth in Rooms and Halls," presented at the 93rd Convention of the Audio Engineering Society, *J. Audio Eng. Soc. (Abstracts)*, vol. 40, p. 1064 (1992 Dec.), preprint 3383.

[7] M. Schroeder, "Natural Sounding Artificial Reverberation," *J. Audio Eng. Soc.*, vol. 10, p. 219 (1962 July).

[8] J. M. Jot and A. Chaigne, "Digital Delay Networks for Designing Artificial Reverberators," presented at the 90th Convention of the Audio Engineering Society, *J. Audio Eng. Soc. (Abstracts)*, vol. 39, p. 383 (1991 May), preprint 3030.

[9] J. O. Smith, "Elimination of Limit Cycles and Overflow Oscillations in Time-Varying Lattice and Ladder Digital Filters," in *Music Applications of Digital Waveguides*, Rep. STAN-M-39, Center for Computer Research in Music and Acoustics (CCRMA), Dept. of Music, Stanford University, Stanford, CA (1987 May).

[10] A. V. Oppenheim, Ed., *Applications of Digital Signal Processing* (Prentice-Hall, Englewood Cliffs, NJ, 1978).

[11] L. B. Jackson, *Digital Filters and Signal Processing*, 3rd ed. (Kluwer Academic, Norwell, MA, 1996).

[12] J. Dattorro, "The Implementation of Recursive Digital Filters for High-Fidelity Audio," *J. Audio Eng. Soc.*, vol. 36, pp. 851–878 (1988 Nov.); Comments, ibid. (*Letters to the Editor*), vol. 37, p. 486 (1989 June); Comments, ibid. (*Letters to the Editor*), vol. 38, pp. 149–151 (1990 Mar.).

[13] P. A. Regalia and S. K. Mitra, "Tunable Digital Frequency Response Equalization Filters," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-35, pp. 118–120 (1987 Jan.).

[14] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing* (Prentice-Hall, Englewood Cliffs, NJ, 1989).

[15] J. Strawn, Ed., *Digital Audio Signal Processing* (A-R Editions, Madison, WI, 1985).

[16] J. A. Moorer, "The Manifold Joys of Conformal Mapping: Applications to Digital Filtering in the Studio," *J. Audio Eng. Soc.*, vol. 31, pp. 826–841 (1983 Nov.).

[17] D. P. Rossum, "Dynamic Digital IIR Audio Filter and Method which Provides Dynamic Digital Filtering for Audio Signals," U.S. patent 5,170,369 (1992 Dec.).

[18] T. I. Laakso, "Error Feedback for Reduction of Quantization Errors due to Arithmetic Operations in Recursive Digital Filters," D. Tech. thesis, Rep. 9, Laboratory of Signal Processing and Computer Technology, Helsinki University of Technology, Espoo, Finland (1991).

[19] K. Steiglitz, *A Digital Signal Processing Primer* (Addison-Wesley, Menlo Park, CA, 1996).

[20] *Moog Voltage-Controlled Ladder Filter*, Internet: http://www.sara.nl/Rick.Jansen/Emusic/Moog/moogvcf_schematic.gif and Internet: http://www-ccrma.stanford.edu/~stilti/papers/moogvcf.pdf.

[21] D. Rossum, "Making Digital Filters Sound 'Analog,'" in *Proc. Int. Computer Music Conf.* (San Jose, CA, 1992), pp. 30–33.

[22] "CEM3328 Four Pole Low Pass VCF," Curtis Electromusic Specialties, Los Gatos, CA (1983).

[23] H. Chamberlin, *Musical Applications of Microprocessors* (Hayden, Indianapolis, IN, 1980).

[24] *Mathematica*, version 2, Wolfram Research, Champaign, IL (1994).

[25] B. L. Evans, L. J. Karam, K. A. West, and J. H. McClellan, "Learning Signals and Systems with *Mathematica*," *IEEE Trans. Education*, vol. 36, pp. 72–78 (1993 Feb.).

[26] K. W. Martin and M. T. Sun, "Adaptive Filters Suitable for Real-Time Spectral Analysis," *IEEE Trans. Circuits Sys.*, vol. CAS-33, pp. 218–229 (1986 Feb.); also *IEEE J. Solid-State Circuits*, vol. SC-21, no. 1 (1986 Feb.), pp. 108–119.

[27] M. R. Petraglia, S. K. Mitra, and J. Szczupak, "Adaptive Sinusoid Detection Using IIR Notch Filters and Multirate Techniques," *IEEE Trans. Circuits Sys. II*, vol. 41, pp. 709–717 (1994 Nov.).

[28] T. Kwan and K. Martin, "Adaptive Detection and Enhancement of Multiple Sinusoids Using a Cascade IIR Filter," *IEEE Trans. Circuits Sys.*, vol. 36, pp. 937–947 (1989 July).

[29] P. P. Vaidyanathan, *Multirate Systems and Filter Banks* (Prentice-Hall, Englewood Cliffs, NJ, 1993).

[30] T. I. Laakso, P. S. R. Diniz, I. Hartimo, and T. C. Macedo, Jr., "Elimination of Zero-Input and Constant-Input Limit Cycles in Single-Quantizer Recursive Filter Structures," *IEEE Trans. Circuits Sys. II*, vol. 39, pp. 638–646 (1992 Sept.).

[31] A. Luthra, "Extension of Parseval's Relation to Nonuniform Sampling," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 36 (1988 Dec.), pp. 1909–1911.

[32] R. W. Schafer and L. R. Rabiner, "A Digital Signal Processing Approach to Interpolation," *Proc. IEEE*, vol. 61, pp. 692–702 (1973 June).

[33] F. R. Moore, "Table Lookup Noise for Sinusoidal Digital Oscillators," *Computer Music J.*, vol. 1, pp. 26–29 (1977 Apr.), also in *Elements of Computer Music* (Prentice-Hall, Englewood Cliffs, NJ, 1990), chap. 3.

[34] R. E. Crochiere and L. R. Rabiner, *Multirate Digital Signal Processing* (Prentice-Hall, Englewood Cliffs, NJ, 1983).

[35] M. Renfors and T. Saramäki, "Recursive *N*th-Band Digital Filters—Parts I and II," *IEEE Trans. Circuits Sys.*, vol. CAS-34, pp. 24–51 (1987 Jan.).

[36] D. Rossum, "An Analysis of Pitch Shifting Algorithms," presented at the 87th Convention of the Audio Engineering Society, *J. Audio Eng. Soc. (Abstracts)*, vol. 37, p. 1072 (1989 Dec.), preprint 2843.

[37] R. Adams and T. Kwan, "Theory and VLSI Architectures for Asynchronous Sample-Rate Converters," *J. Audio Eng. Soc.*, vol. 41, pp. 539–555 (1993 July/Aug.).

[38] T. A. Ramstad, "Digital Methods for Conversion between Arbitrary Sampling Frequencies," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-32, pp. 577–591 (1984 June).

[39] T. I. Laakso, V. Välimäki, M. Karjalainen, and U. K. Laine, "Splitting the Unit Delay—Tools for Fractional Delay Filter Design," *IEEE Signal Process. Mag.*, vol. 13, pp. 30–60 (1996 Jan.).

[40] J. Dattorro, "The Implementation of Digital Filters for High Fidelity Audio, Part II—FIR," in *Audio in Digital Times, Proc. Audio Engineering Society 7th Int. Conf.* (Toronto, ON, Canada, 1989 May 14–17), pp. 168–180.

[41] J. C. Dattorro, A. J. Charpentier, and D. C. Andreas, "Decimation Filter as for a Sigma–Delta Analog-to-Digital Converter," U.S. patent 5,027,306 (1991 June 25).

[42] D. C. Andreas, "VLSI Implementation of a One-Stage 64:1 FIR Decimator," presented at the 89th Convention of the Audio Engineering Society, *J. Audio Eng. Soc. (Abstracts)*, vol. 38, p. 872 (1990 Nov.), preprint 2976.

[43] J. O. Smith and B. Friedlander, "Adaptive, Interpolated Time-Delay Estimation," *IEEE Trans. Aerospace Electron. Sys.*, vol. 21, pp. 180–199 (1985 Mar.).

[44] J. O. Smith III, "Techniques for Digital Filter Design and System Identification with Application to the Violin," Rep. STAN-M-14, Center for Computer Research in Music and Acoustics (CCRMA), Dept. of Music, Stanford University, Stanford, CA (1983 June).

[45] V. Välimäki, T. I. Laakso, and J. Mackenzie, "Elimination of Transients in Time-Varying Allpass Fractional Delay Filters with Application to Digital Waveguide Modeling," in *Proc. Int. Computer Music Conf.* (Banff, AB, Canada, 1995), pp. 327–334.

[46] J. O. Smith, "An Allpass Approach to Digital Phasing and Flanging," Rep. STAN-M-21, Center for Computer Research in Music and Acoustics (CCRMA), Dept. of Music, Stanford University, Stanford, CA (1982 Spring).

[47] W. M. Hartmann, "Flanging and Phasers," *J. Audio Eng. Soc. (Engineering Reports)*, vol. 26, pp. 439–443 (1978 June).

[48] M. L. Beigel, "A Digital 'Phase Shifter' for Musical Applications, Using the Bell Labs (Alles–Fischer) Digital Filter Module," *J. Audio Eng. Soc. (Engineering Reports)*, vol. 27, pp. 673–676 (1979 Sept.).

[49] *DSP56000/DSP56001 Digital Signal Processor User's Manual*, rev. 2 (Motorola, DSP Division, Austin, TX, 1990).

[50] F. F. Lee, "Time Compression and Expansion of Speech by the Sampling Method," *J. Audio Eng. Soc.*, vol. 20, pp. 738–742 (1972 Nov.); also in *Speech Enhancement*, J. S. Lim. Ed. (Prentice-Hall, Englewood Cliffs, NJ, 1983), pp. 286–290.

[51] J. Dattorro, "Using Digital Signal Processor Chips in a Stereo Audio Time Compressor/Expander," presented at the 83rd Convention of the Audio Engineering Society, *J. Audio Eng. Soc. (Abstracts)*, vol. 35, p. 1062 (1987 Dec.), preprint 2500.

The biography for Jon Dattorro was published in the 1997 September issue of the *Journal*.