Fractional Delay Filters—Design and Applications

Chapter · January 2001

DOI: 10.1007/978-1-4615-1229-5_20

2 authors:

Vesa Välimäki
Aalto University

Timo I. Laakso

Chapter 20

Fractional Delay Filters

Design and Applications

Vesa Välimäki[1] and Timo I. Laakso[2]

[1]*Helsinki University of Technology, Laboratory of Acoustics and Audio Signal Processing, Espoo, Finland (e-mail: vesa.valimaki@hut.fi)*

[2]*Helsinki University of Technology, Signal Processing Laboratory, Espoo, Finland (e-mail: timo.laakso@hut.fi)*

Abstract:

In numerous applications, such as communications, audio and music technology, speech coding and synthesis, antenna and transducer arrays, and time delay estimation, not only the sampling frequency but the actual sampling instants are of crucial importance. Digital fractional delay (FD) filters provide a useful building block that can be used for fine-tuning the sampling instants, i.e., implement the required bandlimited interpolation. The FD filters can be designed and implemented flexibly using various established techniques that suit best for the particular application. In this review article, the generic problem of designing digital filters to approximate a fractional delay is addressed. We compare FIR and allpass filter approaches to FD approximation. Time- and frequency-domain characteristics of various

designs are shown to illustrate the nature of different approaches. Special attention is paid to time-varying FD filters and the elimination of induced transients. Also, nonuniform signal reconstruction using polynomial filtering techniques is discussed. Several applications, ranging from synchronization in digital communications to music synthesis, are described in detail. An extensive list of references is provided.

1. INTRODUCTION

The sampling rate must satisfy the Nyquist criterion in order for a sample set to represent adequately the original continuous signal. This problem has been addressed in the sampling theory literature [55], [84], [81]. However, the appropriate sampling rate alone is not sufficient for many applications—also the *sampling instants* must be properly selected. For example, in digital communications, the decisions of the received bit or symbol value are made based on samples of the received continuous-time pulse sequence which should be taken exactly at the middle of each pulse to minimize probability of erroneous decision. This requires that both the sampling frequency and the sampling instants must be synchronized to the incoming signal.

Another class of problems is modeling of dynamical physical systems which usually involves discretization of complex sets of differential equations. Particularly in multidimensional problems, this results in massive computations where controlling the accuracy of the result may be hard. Specifically, problems are caused by exact fitting of the boundary constraints e.g., in simulating an acoustical tube that is precisely as long as required or a vibrating membrane that has exactly circular shape of given diameter, regardless of the employed sampling grid.

Both examples are typical applications of *fractional delay (FD) filters*, i.e., situations where uniform sampling is used and interpolation between samples is required. Fractional delay means, assuming uniform sampling, a delay that is a noninteger multiple of the sample interval. Employing fractional delay filters facilitates the use of traditional well-known methods developed for uniformly sampled signals and yet the observation of signal values at arbitrary locations between the existing samples. Thus, FD filters can be viewed as a first step towards nonuniformly sampled discrete-time systems.

One may claim that this is nothing new: interpolation methods have been known for centuries. Nevertheless, the FD filters have proven useful in providing a systematic framework for solving different interpolation problems and in providing an engineering building block whose properties can be controlled in a desired way.

In this chapter, we review the theory and applications of fractional delay filters. First we introduce the FD filter approximation problem and survey the known techniques for designing nonrecursive (FIR) and recursive (IIR, especially allpass) filters approximating a given FD value. We focus on frequency-domain design methods. This has the advantage of enabling the use of the well-developed toolbox of linear filter design algorithms which often are specified in the frequency domain. Special attention is paid to the implementation of time-varying FD filters and to the elimination of transients in time-varying recursive FD filters.

The filtering of nonuniformly sampled signals is also briefly addressed. The simultaneous reconstruction and noise suppression using polynomial filtering techniques is discussed with examples.

Finally, we proceed to applications. Several fields are discussed where FD filters have been found useful. A selection of applications is considered in more detail, e.g., synchronization of digital modems, conversion between arbitrary sampling frequencies, and simulation of time-varying acoustic propagation delay in virtual audio environments. An extensive list of references is provided.

2. IDEAL FRACTIONAL DELAY

We start by investigating the properties of continuous and discrete-time ideal fractional delay systems. We concentrate on processing of one-dimensional signals that are functions of time.

2.1 Continuous-Time System for Arbitrary Delay

Consider a delay element, i.e., a linear system whose purpose is to delay an incoming continuous-time signal $x_c(t)$ by τ (in seconds). Here the subscript ‘$_c$’ refers to ‘continuous-time’. The output signal $y_c(t)$ of this system can be expressed as

$$y_c(t) = x_c(t-\tau)\,. \tag{1}$$

Additional insight can be gained by considering the delay in the frequency domain. The Fourier transform $X_c(F)$ of signal $x_c(t)$ is defined as

$$X_c(F) = \int_{-\infty}^{\infty} x_c(t) e^{-j2\pi Ft}\, dt\,. \tag{2}$$

where F is frequency in Hz. The Fourier transform $Y_c(F)$ of the delayed signal $y_c(t)$ can be written in terms of $X_c(F)$ as

$$Y_c(F) = \int_{-\infty}^{\infty} y_c(t) e^{-j2\pi Ft}\, dt = \int_{-\infty}^{\infty} x_c(t-\tau) e^{-j2\pi Ft}\, dt = e^{-j2\pi F\tau} X_c(F)\,. \tag{3}$$

The frequency response $H_{\mathrm{id}}(F)$ of the delay element can be expressed by means of Fourier transforms $X_{\mathrm{c}}(F)$ and $Y_{\mathrm{c}}(F)$. This yields

$$H_{\mathrm{id}}(F) = \frac{Y_{\mathrm{c}}(F)}{X_{\mathrm{c}}(F)} = e^{-j2\pi F\tau}. \tag{4}$$

Hence, in the frequency domain the delay τ corresponds to a complex exponential factor $e^{-j2\pi F\tau}$. In some applications, it is desired to approximate a given delay directly in the continuous-time domain by analog filters [114]. However, in this chapter we focus on discrete-time delay systems only.

2.2 Discrete-Time System for Arbitrary Delay

If the Fourier transform $X_{\mathrm{c}}(F)$ is non-zero only on a finite interval $-W \le F \le W$, the continuous-time signal $x_{\mathrm{c}}(t)$ is said to be *bandlimited*. According to the sampling theorem, signal $x_{\mathrm{c}}(t)$ may then be expressed by its samples $x(n) = x_{\mathrm{c}}(nT)$, where n is the sample index (n = ..., –1, 0, 1, 2, …) and $T = 1/F_{\mathrm{s}}$ is the sampling interval when the sampling frequency is $F_{\mathrm{s}} > 2W$. For simplicity, we omit T and use $x(n)$ to denote the samples of the discrete-time signal.

We want to express the *discrete-time version* of the delay operation for a sampled bandlimited signal. The outcoming discrete-time signal $y(n)$ would ideally be written as

$$y(n) = x(n - D), \tag{5}$$

where $D = \tau / T$ is the desired delay normalized with respect to the sampling interval. Note that τ / T is generally irrational since τ is usually not an integral multiple of sampling interval T. The delay D may thus be written in the form $D = \text{floor}(D) + d$, where $0 \le d < 1$ is the *fractional delay* and the floor function returns the greatest integer less than or equal to D.

Unfortunately, as $x(n)$ and $y(n)$ are sequences whose values are defined for integer argument values only, equation (5) is meaningful only for integral values of D. Then the samples of the output sequence $y(n)$ are equal to the delayed samples of the input sequence $x(n)$, and the delay element is called a *digital delay line*. However, if D were real-valued, the delay operation would not be this simple, since the output value would lie somewhere between the existing samples of $x(n)$. In this case, the sample values of $y(n)$ have to generated based on the sequence $x(n)$ by using *interpolation*. This is known as the fractional delay problem in digital signal processing.

The spectrum of a discrete-time signal can be expressed by means of the *discrete-time Fourier transform* (DTFT). In this integral transform, the time variable is discretized, but the frequency variable is continuous. The DTFT of signal $x(n)$ is defined as (see, e.g., [53], pp. 140–151)

$$X(e^{j2\pi f}) = \sum_{n=-\infty}^{\infty} x(n)e^{-j2\pi fn} \ , \tag{6}$$

where f is the normalized frequency, $f = F/F_s$, and thus $f = 1/2$ corresponds to the Nyquist limit. The frequency response of the ideal discrete-time delay element can be given as

$$H_{\text{id}}(e^{j2\pi f}) = \frac{Y(e^{j2\pi f})}{X(e^{j2\pi f})} = e^{-j2\pi fD} \, . \tag{7}$$

This result is comparable to (4)—only now the frequency response is periodic due to discretization in time. To be consistent with the z-transform notation used commonly in digital signal processing literature, we express the transfer function as

$$H_{\text{id}}(z) = \frac{Y(z)}{X(z)} = \frac{z^{-D}X(z)}{X(z)} = z^{-D} \, , \tag{8}$$

where D is the delay in samples. Note that the z-transform representation in (8) is used in the Fourier transform sense so that $z = e^{j2\pi f}$. In principle, the z-transform is defined only for integral powers of z and thus, if D were real-valued, the term z^{-D} should be written as an infinite series making the notation unnecessarily involved.

To understand how to produce a fractional delay using a discrete-time system, it is necessary to discuss interpolation techniques. Interpolation of a discrete-time signal is based on the fact that the amplitude of the corresponding continuous-time *bandlimited* signal changes smoothly between the sampling instants.

2.3 Fractional Delay and Signal Reconstruction

The fractional delay d can in principle have any value between 0 and 1. Therefore, to produce an arbitrary fractional delay for a discrete-time signal $x(n)$, one needs to know a way to compute the value of the underlying continuous-time signal $x_c(t)$ for all t. This leads us to the problems of sampling and reconstruction. According to the sampling theorem, a uniformly sampled signal can be reconstructed from its samples as follows [55]:

$$x_{\mathrm{c}}(t) = \sum_{n=-\infty}^{\infty} x(n)\mathrm{sinc}\left[F_{\mathrm{s}}(t - nT)\right]. \tag{9}$$

The sinc function is defined as $\mathrm{sinc}(t) = \sin(\pi t) / \pi t$.

According to (9) the ideal bandlimited interpolator has a continuous-time impulse response

$$h_{\mathrm{c}}(t) = \mathrm{sinc}(F_{\mathrm{s}}t). \tag{10}$$

This impulse response converts a discrete-time signal to a continuous-time one.

In delay applications, however, it is necessary to know the value of a signal at a single time instant between the samples. The desired result may be obtained by shifting the impulse response (10) by D and then sampling it at equidistant points. The output $y(n)$ of the ideal discrete-time fractional delay element is computed as

$$y(n) = x(n - D) = \sum_{k=-\infty}^{\infty} x(k)\mathrm{sinc}(n - D - k)\,. \qquad (11)$$

In conclusion, a fractional delay requires reconstruction of the discrete-time signal and shifted resampling of the resulting continuous-time signal. In (11) these two operations have been combined. The impulse response of an ideal discrete-time delay element can be expressed as

$$h_{\mathrm{id}}(n) = \mathrm{sinc}(n - D). \qquad (12)$$

The properties of this ideal filter are reviewed in the following.

2.4 Characteristics of the Ideal Fractional Delay Element

The *magnitude response* of the ideal fractional delay element can be obtained from the frequency response (7) as

$$\left|H_{\mathrm{id}}(e^{j2\pi f})\right| = 1\,. \qquad (13)$$

Thus, the magnitude response of the ideal delay element is flat. Its *phase response* is $\theta_{id}(f) = -2\pi fD$. Its *phase delay* and *group delay* are, respectively,

$$\tau_{p,id}(f) = -\frac{\theta_{id}(f)}{2\pi f} = D \tag{14}$$

and

$$\tau_{g,id}(f) = -\frac{1}{2\pi}\frac{d\theta_{id}(f)}{df} = D\,. \tag{15}$$

Both the phase delay and the group delay are measures for the delay of the system. Their difference is typically illustrated by considering an amplitude-modulated signal for which the phase delay and the group delay describe the delay experienced by the carrier signal and the envelope, respectively [110]. For a linear-phase system, the measures yield an identical result, which is independent of frequency—just like in (14) and (15).

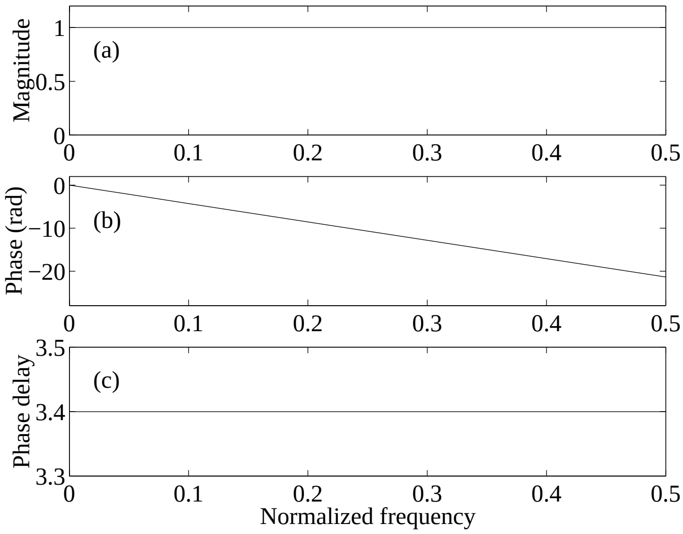

Fig. 1. Frequency-domain characteristics of an ideal FD element for D = 3.4 samples: (a) magnitude response, (b) phase response, and (c) phase delay.

Fig. 1 illustrates the characteristics of an ideal FD transfer function. The responses are shown up to the Nyquist frequency, which is equivalent to $F_s/2$, or normalized frequency f = 1/2. The flat magnitude response, the linear phase response, and the constant phase delay (equivalent to the group delay in this case) are characteristics of a linear-phase allpass system. However, a fractional-delay element is not a linear-phase system in the traditional sense[†] since the impulse response (12) is generally not symmetric. Oppenheim and Schafer (see [108], Sec. 5.7) use the term “generalized linear phase” in the context of fractional delay filters.

If D is an integer ($d = 0$), the impulse response of the delay element (12) is zero at all sampling points except at $n = D$. An example is presented in Fig. 2(a) where both the continuous-time and the sampled impulse response of the ideal FD filter is shown. In this case, a chain of unit delays, as discussed earlier, can implement the delay element.

When D is a fractional number, i.e., $0 < d < 1$, the impulse response has non-zero values at all index values n. Figure 2(b) gives an example where $D = 3.4$ samples. In this case, the impulse response $h_{\mathrm{id}}(n)$ of the discrete-time delay element, which is a shifted and sampled version of the sinc function, is infinitely long in both directions. For this reason, the impulse response corresponds to a *noncausal* filter which cannot be made causal by a finite shift in time. In addition, the filter is not stable since the impulse response (12) is not absolutely summable. The ideal filter is thus *nonrealizable*. To produce a realizable fractional delay filter, some finite-length, causal approximation for the sinc function has to be used.

---

† Conventional linear-phase FIR filters are known to have a symmetric or anti-symmetric impulse response with respect to its midpoint (see, e.g., [108] or [53]).

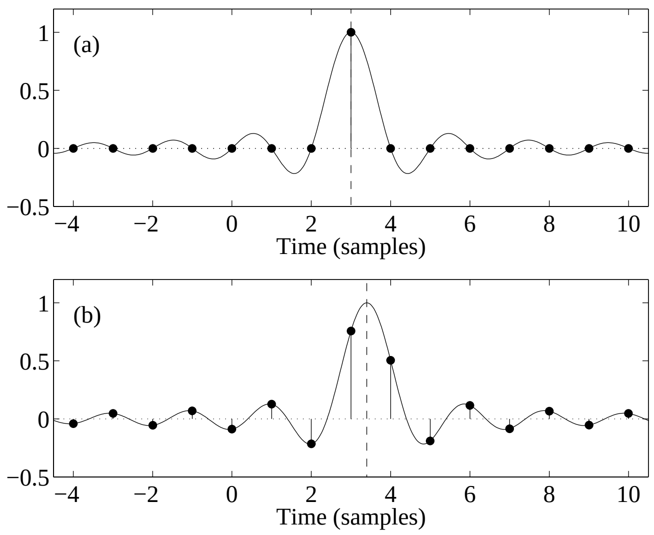

Fig. 2. Continuous-time (solid line) and sampled (l) impulse response of the ideal fractional delay filter, when the delay is (a) D = 3.0 samples and (b) D = 3.4 samples. The vertical dashed line indicates the midpoint of the continuous-time impulse response in each case.

Before considering the approximations, we notify a particular property of the ideal frequency response that makes the FD approximation difficult. The imaginary part of the ideal transfer function (7) at the Nyquist frequency (f = 1/2) is $-\sin(\pi D)$. This implies that when D has a non-zero fractional part d, the transfer function has a complex value at f = 1/2. However, discrete-time filters with real coefficients have the property that their frequency response is real-valued at f = 1/2. Thus, the approximation error cannot be smaller than $|\sin(\pi D)|$ at the Nyquist frequency [161], [12]. This strongly influences the behavior of FD filters and facilitates understanding of design obligations.

3. DESIGN OF FIR FRACTIONAL DELAY FILTERS

As discussed in the previous section, the ideal FD filter has an infinitely long impulse response which is not realizable in practical systems. In this section, we focus on the approximation of the fractional delay using realizable filters, the simplest class of which are finite impulse response (FIR) filters. For a more detailed discussion and examples on FIR FD filter design, see [12] and [71].

The transfer function of an FIR filter is of the form

$$H(z) = \sum_{n=0}^{N} h(n) z^{-n} , \tag{16}$$

where N is the order of the filter, and $h(n)$ (n = 0, 1, ..., N) are the real-valued coefficients that form the impulse response of the FIR filter. Note that the length of the impulse response, i.e., the number of the filter coefficients, is $L = N + 1$. The block diagram of the FIR filter is shown in Fig. 3.

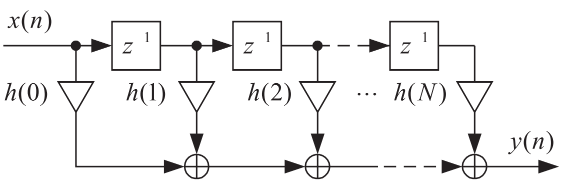

Fig. 3. Block diagram of an FIR filter of order *N*.

In the design procedure our aim is to minimize the error function which is defined as the difference of the ideal frequency response and the approximation, i.e., the frequency-response error

$$E(e^{j2\pi f}) = H_{\mathrm{id}}(e^{j2\pi f}) - H(e^{j2\pi f}). \tag{17}$$

The choice of error norm affects the choice of the design method. In the following, we consider the three most common approximation criteria:

1. minimization of the L_2 *norm* of the error, or the error power,
2. the *maximally-flat* criterion, and
3. minimization of the L_∞ *norm* of the error (minimax or Chebyshev criterion).

However, before going into details, let us discuss the important concept of polyphase filters and their relation to FD approximation.

3.1 Polyphase FIR Filters

The polyphase filter structure has been developed for multirate signal processing. It is a straightforward way to implement interpolation and decimation filters. It also offers an easy method to design FIR fractional delay filters using optimization techniques for linear-phase FIR filters [18], [7], [96], [19], [131], [172], [124], [133]. This method only allows the design of

FD filters for rational fractions of a unit delay. For example, to split a unit delay into Q steps, one can design a Qth-band lowpass filter with the normalized bandwidth of $1/2Q$ and form the Q-branch polyphase structure by picking up every Qth sample value from the filter. Each of these branches approximates a fractional delay d_k so that

$$d_k = \frac{k}{Q}, \tag{18}$$

where $k = 0, 1, 2, ..., Q - 1$. Thus, each polyphase branch can be used as a fractional delay filter. In order to achieve a comparable frequency response for every branch, the length of the prototype filter should be a multiple of Q plus 1.

For example, if it is desired to divide a sampling interval into 4 steps, i.e., $Q = 4$, and the length of each filter is chosen to be $L = 10$, the prototype filter to be designed should be of length $QL + 1 = 41$. An example of the impulse response of a linear-phase prototype filter of this length is shown in Fig. 4, and Fig. 5 presents the four FD filters obtained from the prototype. Note that the filter of Fig. 5(c) approximating the delay $d = 0.5$ has a symmetric impulse response, which implies that it is a linear-phase FIR filter. The impulse responses in Fig. 5(b) and (d) are time-reversed versions of each other.

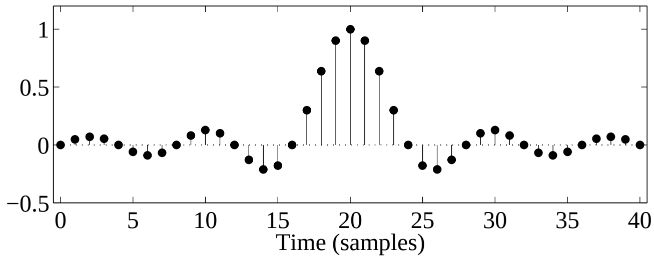

Fig. 4. Prototype linear-phase Qth-band FIR filter used for creating a polyphase structure (Q = 4, L = 10).

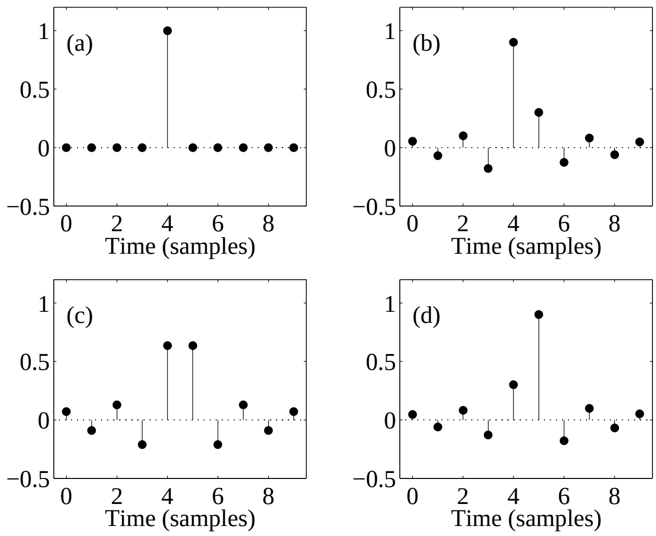

Fig. 5. Impulse responses (length L = 10) of four fractional delay filters obtained by collecting every fourth sample value with different offsets from the prototype impulse response of Fig. 4. The corresponding fractional delay values are (a) 0.0, (b) 0.25, (c) 0.5, and (d) 0.75.

The lowpass prototype filter should be linear-phase, but it can be designed with any available method. However, the optimality of the prototype filter is not shared with the branch filters. For example, equiripple magnitude characteristic will be lost. A feature that limits the usefulness of this approach is that only rational fractional delays are available. Naturally, one can obtain an arbitrarily good resolution of fractional delays by making Q large but this leads to very high-order FIR filter design. Since there are many other methods available for fractional delay filter design, the polyphase approach seems outdated for this purpose. However, rather than for approximation, the multirate approach can be used for implementing high-quality wideband FD filters, as proposed in [100]. Using a polyphase implementation with two branches, the accuracy of approximation can be increased without excessive total processing delay.

3.2 Least Squared Integral Error Design

The intuitively most attractive method for designing realizable FD filters for arbitrary delay values is the truncation of the ideal impulse response defined by (12). This method minimizes the least squared (LS) error function E_{LS} which is equal to the L_2 norm (integrated squared magnitude) of the frequency-response error (17), i.e.,

$$E_{\mathrm{LS}} = 2\int_0^{1/2} \left| E(e^{j2\pi f}) \right|^2 df = 2\int_0^{1/2} \left| H_{\mathrm{id}}(e^{j2\pi f}) - H(e^{j2\pi f}) \right|^2 df \ . \tag{19}$$

Using Parseval’s relation, we get

$$E_{\mathrm{LS}} = \sum_{n=-\infty}^{\infty} \left[h_{\mathrm{id}}(n) - h(n) \right]^2 = \sum_{n=-\infty}^{\infty} \left[h_{\mathrm{id}}^2(n) + h^2(n) - 2h(n)h_{\mathrm{id}}(n) \right]. \qquad (20)$$

From (20), it is possible to derive a closed-form solution for the squared integral error in the case of fractional delay approximation. According to Parseval's relation, the total sum of the ideal impulse response is equal to 1. The second and third term of (20) include the coefficients of the Nth-order FIR filter and thus the summation indices can be limited. The closed-form solution is

$$E_{\mathrm{LS}} = 1 + \sum_{n=0}^{N} \left[h^2(n) - 2h(n)\mathrm{sinc}(n - D) \right]. \qquad (21)$$

The optimal solution for an Nth-order FIR FD filter in the L_2 sense is the one with $N + 1$ coefficients truncated symmetrically around the maximum value, i.e., the central point of $h_{\mathrm{id}}(n)$. Truncating the shifted sinc function is an easy way to design FIR FD filters. This approach has been proposed, e.g., by Sivanand *et al.* [147], [148] and Cain *et al.* [12]. However, it is often not useful since truncation of the impulse response introduces ripple in the frequency response. This is called the *Gibbs phenomenon* (see, e.g., [53]). It causes the maximum deviation from the ideal frequency response to remain approximately constant irrespective of the filter order. This applies to both the magnitude and the phase response.

We present an example where the impulse response of the ideal FD filter for 3.4 samples—see Fig. 2(b)—has been used as a prototype. Eight coefficient values, $h_{id}(0)$, $h_{id}(1)$, $h_{id}(2)$, ..., and $h_{id}(7)$, have been truncated symmetrically about the midpoint. The impulse response of the resulting filter is displayed in Fig. 6(a). These are the coefficients of an LS FIR filter of order $N = 7$ (or, length $L = 8$), which approximates the delay $D = 3.4$ samples. The magnitude of the frequency-response error $E(e^{j2\pi f})$ computed using (17) is presented in Fig. 6(b). Note how the local maxima of the error function increase towards high frequencies.

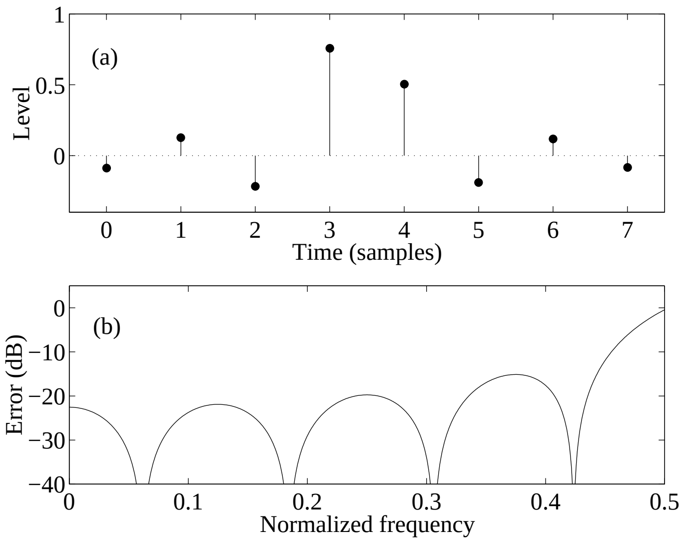

Fig. 6. (a) Impulse response and (b) frequency-response error of the LS FIR FD filter obtained by truncating the impulse response of the ideal FD filter ($N = 7$, $D = 3.4$) shown in Fig. 2(b).

Rather than the error function itself, engineers often find it more useful to examine the magnitude response and the phase delay or group delay of the filter. The magnitude response of the above filter is shown in Fig. 7(a). The ripple caused by the Gibbs phenomenon can be seen: The oscillation of the response grows towards higher frequencies, and the phase-delay response shown in Fig. 7(b) exhibits similar behavior. Note that the phase delay curve reaches 3.0 samples at the Nyquist frequency (i.e., normalized frequency 0.5).

A variation of the LS design technique is to use a *lowpass* interpolator as a prototype filter instead of the fullband fractional delay filter [71]. The Gibbs phenomenon will be reduced considerably, as desired, but the usable bandwidth will contract as well. Another variation of the basic LS design is to use a reduced bandwidth with a *smooth transition band* function (see, e.g., [112], pp. 63–70). This will make the impulse response of the FD element decay fast. The impulse response will still be infinitely long and must be truncated, but the Gibbs phenomenon is guaranteed to be reduced, since the discontinuity is not as sharp as originally. A good choice for the transition band is a low-order spline multiplied by $e^{-j2\pi fD}$ [71].

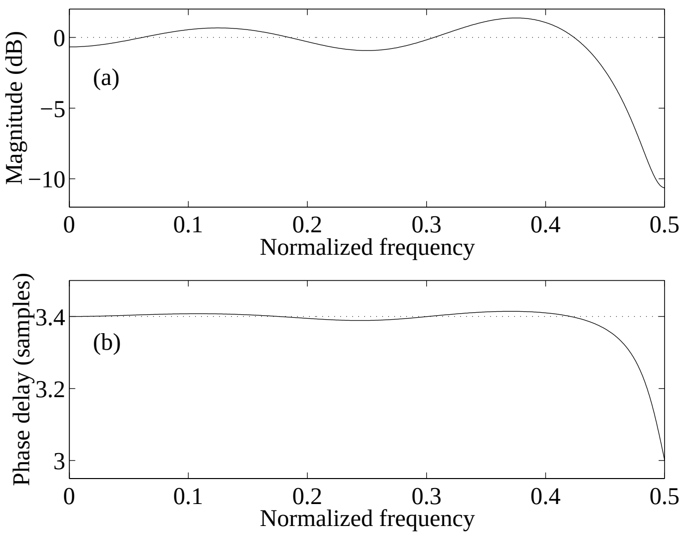

Fig. 7. (a) Magnitude response (solid line) and (b) phase delay (solid line) of the LS FIR FD filter of Fig. 6(a). The dotted lines indicate the corresponding ideal characteristics.

A well-known method to reduce the Gibbs phenomenon in FIR filter design is to multiply the ideal impulse response with a bell-shaped non-negative finite-length weighting function, i.e., a window function (see, e.g., [112], pp. 71–83). Since the truncation yields the optimal solution in the LS sense, any windowing method is bound to be worse. Thus, the window-based design method does not minimize the LS error measure, but it is an *ad hoc* modification of the LS technique.

The impulse response of an FIR filter designed by the windowed LS method can be written in the form

$$h(n) = \begin{cases} w(n-D)\mathrm{sinc}(n-D) & \text{for } 0 \le n \le N \\ 0 & \text{otherwise.} \end{cases} \tag{22}$$

Note that the midpoint of the window function $w(n)$ of length $N + 1$ has been shifted by D so that the shifted sinc function will be windowed symmetrically with respect to its center. The window function $w(n - D)$ is generally *asymmetric*, however.

Many window functions, such as the Hamming and Hanning windows, can be easily delayed by a fractional value D [71], [12], [13], whereas Dolph–Chebyshev or Saramäki windows are defined in the frequency domain and only approximative shifting techniques are known [67]. Special window functions for FD applications have been developed by Cain and Yardim *et al.* [13], [195], [196].

3.3 Weighted Least Squared Integral Error FIR Approximation of a Complex Frequency Response

In principle, the FIR fractional delay filter with the smallest LS error in the defined approximation band is accomplished by defining the response only in that part of the frequency band and by leaving the rest out of the error measure as a “don’t care” band [71]. This scheme also enables frequency-domain weighting of the LS error. This technique minimizes the following error function

$$E_{\mathrm{GLS}} = 2\int_0^{1/2} W(f)\left|E(e^{j2\pi f})\right|^2 df = 2\int_0^{1/2} W(f)\left|H(e^{j2\pi f}) - H_{\mathrm{id}}(e^{j2\pi f})\right|^2 df \ , \tag{23}$$

where the error is defined in the lowpass frequency band [0, $F_s/2$] only, and $W(f)$ is the nonnegative frequency-domain weighting function (not to be confused with the time-domain window function $w(n)$).

The optimal solution can be obtained by solving a set of $N + 1$ linear equations, as presented in [71]. Numerical problems may arise, particularly in narrowband approximation [71]. However, in FD filter design this is not typical. If the weighting function is $W(f) = 1$ for $f \leq \alpha$ and $W(f) = 0$ for $f > \alpha$, where $0 < \alpha \leq 1$, the solution can be given in closed form [174]. Also the effect of the input signal spectrum can be taken into account: Oetken *et al.* [106] proposed a *stochastic LS approach* to the design of interpolating filters. In this technique, the design criterion is the minimum expected mean squared output error. The design technique enables a great deal of flexibility and it is recommended whenever the filter properties need to be tailor made with care.

3.4 Maximally-Flat FIR FD Design: Lagrange Interpolation

In many applications, it is important that the delay is approximated accurately at low frequencies. This can be attained by setting the error function and its N derivatives to zero at the zero frequency. This is the *maximally-flat* (MF) design at $f = 0$. It is interesting to notice that the FIR filter coefficients obtained by this method are the same as the weighting coefficients in the classical *Lagrange interpolation* for uniformly sampled data:

$$h(n) = \prod_{\substack{k=0 \\ k \neq n}}^{N} \frac{D-k}{n-k} \quad \text{for } n = 0, 1, 2, \ldots, N\,. \tag{24}$$

The first-order (N = 1) Lagrange interpolation is equivalent to *linear interpolation*. The coefficient formulas for low-order Lagrange interpolators are given in Table 1. It can be shown that the impulse response of the Lagrange interpolator (24) converges to the shifted and sampled sinc function (12), the ideal FD filter, as N approaches infinity (for a proof, see [81], pp. 100–102).

Table 1. Coefficients of low-order Lagrange interpolation filters.

	$h(0)$	$h(1)$	$h(2)$	$h(3)$
$N = 1$	$1 - D$	D		
$N = 2$	$(D-1)(D-2)/2$	$-D(D-2)$	$D(D-1)/2$	
$N = 3$	$-(D-1)(D-2)(D-3)/6$	$D(D-2)(D-3)/2$	$-D(D-1)(D-3)/2$	$D(D-1)(D-2)/6$

Lagrange interpolation has been studied in a large number of papers in the signal processing literature up to the present day [142], [156], [55],

[107], [72], [63], [6], [147], [79], [80], [47], [95], [173], [174], [64], [48], [87], [164], [16], [130], [23] [24].

We present a design example with the same design parameters as in Section 3.2 ($N = 7$, $D = 3.4$). The impulse response of the Lagrange interpolator, computed using (24), is shown in Fig. 8(a). The corresponding frequency-response error magnitude, shown in Fig. 8(b), is now very small at low frequencies ($-\infty$ dB at $f = 0$), but smoothly increases in a monotonic fashion. The magnitude and phase-delay responses of the filter—see Fig. 8(c) and (d)—exhibit an increasing deviation from their ideal values, which are obtained at zero frequency.

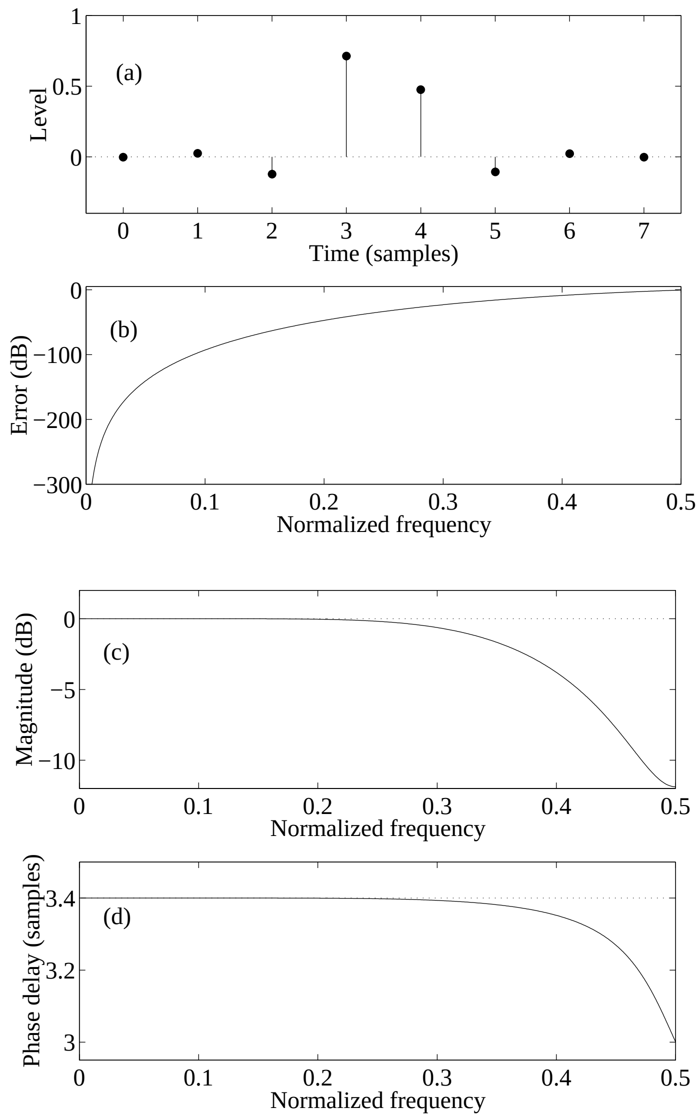

Fig. 8. (a) Impulse response, (b) frequency-response error, (c) magnitude response, and (d) phase delay of a Lagrange interpolating filter (N = 7, D = 3.4). The dotted lines indicate the ideal characteristics in (c) and (d).

3.5 Minimax Design of FIR FD Filters

If it is desired to minimize the peak approximation error, the *minimax*, or *Chebyshev*, error criterion should be used. This approximation problem can usually be solved only by iterative techniques, which are based on either an iterative weighted least squares algorithm or a modification of the Remez exchange algorithm. The minimax optimization problem is rendered particularly demanding since the target frequency response is complex-valued. (Note that the Remez algorithm originally assumes that the function to be approximated is real-valued.) Many advanced algorithms for complex approximation with minimax error characteristics have been presented [112], [117], [144], [77], [58]. The application of various minimax approximation methods to FIR FD filters have been tackled by Oetken [107], Pyfer and Ansari [121], Putnam and Smith [120], and Brandenstein and Unbehauen [11]. For a more detailed discussion on minimax FIR FD filters, see [71]. The method proposed by Oetken is particularly simple [107]. It is based on an odd-order linear-phase Chebyshev prototype filter (d = 0.5), whose zeros are modified with a matrix operation to obtain a fractional delay filter for a given delay parameter d.

Fig. 9 gives an example of FD filter design according to Oetken's method. The delay parameter is the same as in previous examples, D = 3.4.

The length of the filter is 8, and the approximation bandwidth is [0, 0.4F_s]. It is seen that the impulse response is only slightly different from those of other FIR filters. However, the largest peaks of the frequency-response error are now maintained at the level of about –27 dB in the approximation band, i.e., from 0 up to the normalized frequency 0.4. Thus, the equiripple design is better in this sense than the fullband LS FIR filter whose error curve was displayed in Fig. 6(b)—the maximum error of the LS filter in the same frequency band is about –15 dB. The Lagrange interpolating filter is still worse in this kind of comparison: its frequency-response error (see Fig. 8(b)) at normalized frequency 0.4 is only –8.7 dB.

3.6 Fractional Delay Filters Based on Splines

Splines are a class of polynomial interpolation techniques that have several applications in numerical computation. The B-spline interpolator can be implemented as an all-pole or FIR filter. It has been found effective in many DSP applications, such as image processing [52], [171], [26], sampling-rate conversion [14], [20], [204], [205], and signal reconstruction [169], [170]. Aldroubi *et al.* have proved that a cardinal spline interpolator converges toward the ideal interpolator as the order of the filter approaches infinity [4]. The properties of B-spline and Lagrange interpolation have been compared in [78]. A generalization of B-spline interpolators and an efficient implementation structure have been proposed in [30].

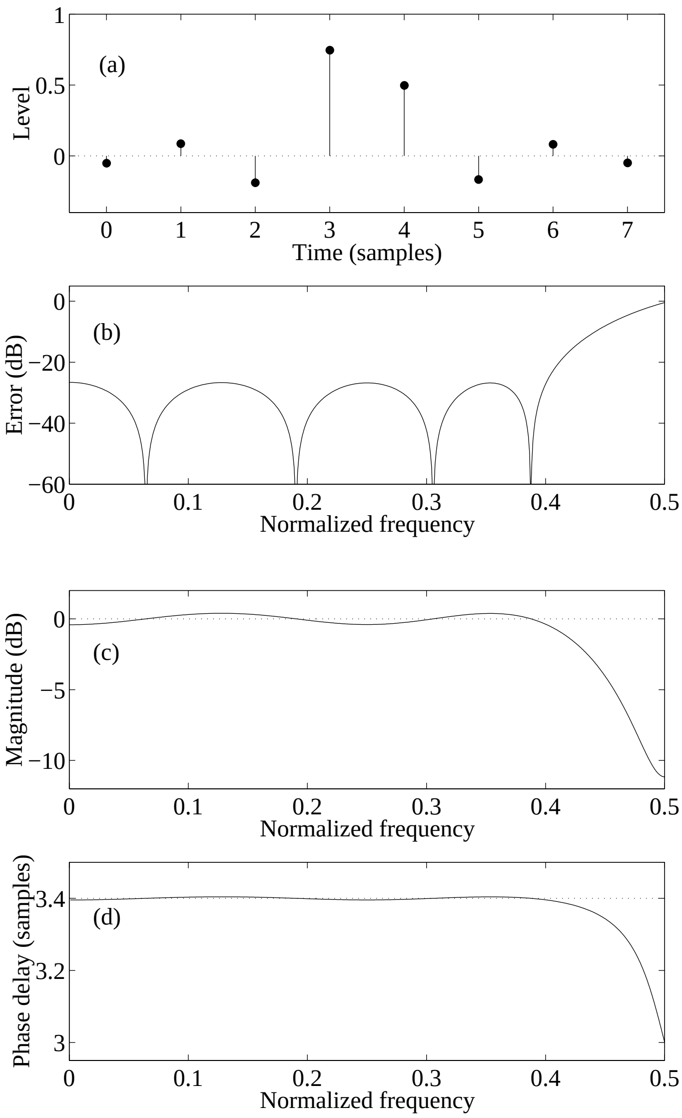

Fig. 9. (a) Impulse response, (b) frequency-response error, (c) magnitude response, and (d) phase delay of an equiripple FIR FD filter designed using Oetken's method (N = 7, D = 3.4). The dotted lines indicate the ideal characteristics in (c) and (d).

3.7 General Properties of FIR FD filters

Some general properties of FIR FD filters can be summarized as follows:

1. The best approximation for a given design method and a given filter order N is obtained when $(N - 1)/2 \leq D < (N + 1)/2$ (see, e.g., [60] or [71])[†]. This implies that the best FD filters are those whose impulse response is almost symmetric. This also means that the total delay of the filter is close to $N/2$ samples.
2. The approximation error decreases as the order N is increased (see, e.g., [142] or [71]). However, it is typical in practical applications that the smallest acceptable filter order N is used to save resources and to keep the total delay small. Thus, it becomes essential to choose the right method to design an FD filter that accomplishes the required task with the smallest possible order N.
3. Since FIR FD filters do not include any feedback, they are always stable. Furthermore, the transients due to abrupt changes in the input signal or due to changes in filter coefficient values are always finite-length. This is important in applications where the FD value needs to be controlled.

[†] In the case of Lagrange interpolation, the consequences of interpolation outside the central interval have been discussed in [142] and [174].

4. DESIGN OF IIR FRACTIONAL DELAY FILTERS

As is well known from the general filter approximation theory, the approximation of a given frequency response with a recursive (IIR) filter achieves the same quality with a lower complexity (e.g., a smaller number of multiplications and additions) than an FIR filter [112]. However, the design of general IIR fractional delay filters having a different numerator and denominator polynomials has not been much discussed in the literature. An example is the study by Tarczynski and Cain where reduced-bandwidth IIR fractional delay filters are optimized iteratively [161]. The design of optimal IIR filters is a difficult problem. One particular difficulty with recursive filters is that they may be unstable whereas FIR filters are always stable. The design procedure has to account for this possibility and ensure the stability of the IIR filter.

Since the magnitude response of an ideal FD element is perfectly flat, we consider here only the so-called *allpass filters*. Their magnitude response is always exactly flat irrespective of the filter coefficients. Allpass filters are typically used for phase equalization and other signal processing tasks where the phase characteristics are of greatest concern. Since the FD approximation is essentially a phase approximation problem, the allpass filter is particularly well suited to this task.

4.1 Discrete-Time Allpass Filter

A discrete-time allpass filter has the transfer function

$$A(z) = \frac{z^{-N} D(z^{-1})}{D(z)} = \frac{a_N + a_{N-1} z^{-1} + \ldots + a_1 z^{-(N-1)} + z^{-N}}{1 + a_1 z^{-1} + \ldots + a_{N-1} z^{-(N-1)} + a_N z^{-N}}\,, \tag{25}$$

where N is the order of the filter, $D(z)$ is the denominator polynomial, and the filter coefficients a_k (k = 1, 2, ..., N) are real-valued. The numerator of the allpass transfer function is a mirrored version of the denominator polynomial. The direct-form I implementation structure of the allpass filter is shown in Fig. 10. The output signal $y(n)$ of the allpass filter is computed using the following difference equation:

$$\begin{aligned} y(n) &= a_N x(n) + a_{N-1} x(n-1) + \ldots + a_1 x(n-N-1) + x(n-N) \\ &\quad - a_1 y(n) - \ldots - a_{N-1} y(n-N+1) - a_N y(n-N) \\ &= a_N \left[x(n) - y(n-N) \right] + a_{N-1} \left[x(n-1) - y(n-N+1) \right] + \ldots \\ &\quad + a_1 \left[x(n-N-1) - y(n) \right] + x(n-N). \end{aligned} \tag{26}$$

The latter form indicates that one output sample of the Nth-order allpass filter can be computed with N multiplications and $2N$ additions (or subtractions).

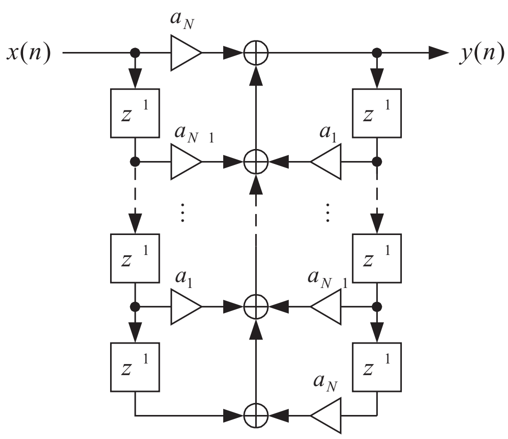

Fig. 10. Direct form I implementation of an *N*th-order allpass filter.

The poles of a stable allpass filter are located inside the unit circle in the complex plane. Since the same coefficients are used in the numerator, its zeros are located outside the unit circle so that their angle is the same but the radius is the inverse of the corresponding pole. For this reason the magnitude response of an allpass filter is flat. It is expressed as

$$\left|A(e^{j2\pi f})\right| = \left|\frac{e^{-j2\pi fN} D(e^{-j2\pi f})}{D(e^{j2\pi f})}\right| = 1\,. \tag{27}$$

Obviously, the name ‘allpass’ filter comes from the above property that this filter passes signal components of all frequencies without attenuating or boosting them. The frequency response of the allpass filter can be written in the form $A(e^{j2\pi f}) = \exp[j\theta_A(f)]$. This form stresses the fact that the main feature of an allpass filter is its *phase response* $\theta_A(f)$, which can be expressed as

$$\theta_A(f) = \arg\left\{A(e^{j2\pi f})\right\} = -2N\pi f + 2\theta_D(f)\,, \tag{28}$$

where $\theta_D(f)$ is the phase response of $1/D(e^{j2\pi f})$, i.e.,

$$\theta_D(f) = \arg\left\{\frac{1}{D(e^{j2\pi f})}\right\} = \mathrm{unwrap}\left[\arctan\left\{\frac{\sum_{k=1}^{N} a_k \sin(2\pi fk)}{1+\sum_{k=1}^{N} a_k \cos(2\pi fk)}\right\}\right], \tag{29}$$

where the ‘unwrap’ function produces the desired monotonically decreasing phase function by subtracting the necessary amount of multiples of π. The *phase delay* $\tau_{p,A}(f)$ of an allpass filter can be expressed according to (28) as

$$\tau_{p,A}(f) = -\frac{\theta_A(f)}{2\pi f} = N - 2\tau_{p,D}(f)\,, \tag{30}$$

where $\tau_{p,D}(f)$ is the phase delay of $1/D(e^{j2\pi f})$. The corresponding *group delay* is given by

$$\tau_{g,A}(f) = -\frac{1}{2\pi}\frac{d\theta_A(f)}{df} = N - 2\tau_{g,D}(f)\,, \tag{31}$$

where $\tau_{g,D}(f)$ is the group delay of $1/D(e^{j2\pi f})$.

4.2 Design of Allpass Fractional Delay Filters

There is a noteworthy difference between the design of FIR and allpass filters: the coefficients of an FIR filter are easily obtained by the inverse discrete-time Fourier transform of the frequency-domain specifications, since the coefficients of an FIR filter are equal to the samples of its impulse response. However, the relationship between the transfer function coefficients and the impulse response of an allpass filter or any other recursive filter is not that simple. Hence, most of the design techniques for allpass filters are iterative. Many design methods for FD allpass filters are counterparts of FIR design methods discussed previously in this chapter.

A time-domain approach to FD approximation using an allpass filter can be obtained with the impulse-response matching technique proposed by Strube [157]. According to our experiments, the technique yields poor results when the truncated impulse response of the ideal FD element (12) is used as the prototype (various integer parts floor(D) < N + 1 were tested). In the following our focus is on frequency-domain design methods.

The desired or ideal phase response in the fractional delay approximation problem is $-2\pi fD$, as can be seen from (7). The phase error of an allpass filter can be defined as the deviation from the desired phase function $\theta_{\mathrm{id}}(f)$ as

$$\Delta\theta(f) = \theta_{\mathrm{id}}(f) - \theta_A(f). \qquad (32)$$

4.2.1 LS Design of Allpass FD Filters

The weighted least-squares *phase* error that is to be minimized is defined as

$$E_{\mathrm{LS}} = 2\int_0^{1/2} W(f)\left|\Delta\theta(f)\right|^2 df \;, \tag{33}$$

where $W(f)$ is a nonnegative weighting function. Lang and Laakso have introduced an iterative algorithm for the design of the allpass filter coefficients [75]. The algorithm typically converges to the desired solution, but this cannot be guaranteed. The stability is not guaranteed either.

The LS *phase delay* error can be defined as

$$E_{\mathrm{LS}} = 2\int_0^{1/2} W(f)\left|\Delta\tau_{\mathrm{p}}(f)\right|^2 df = 2\int_0^{1/2} \frac{W(f)}{(2\pi f)^2}\left|\Delta\theta(f)\right|^2 df \;. \tag{34}$$

In other words, the phase delay error solution is obtained by introducing an additional weighting function $1/(2\pi f)^2$ to the phase error (33). An eigenfilter formulation of this allpass design problem was presented by Nguyen *et al.* [105].

4.2.2 Maximally-Flat Design of Allpass FD Filters

The maximally-flat group delay design is the only known allpass FD filter design technique that has a closed-form solution. Thiran proposed an analytic solution for the coefficients of an *all-pole* lowpass filter with a maximally-flat group delay response at the zero frequency [166]. A drawback of Thiran's design technique is that the magnitude response of the all-pole lowpass filter cannot be controlled. However, the same coefficients can be used also for the numerator polynomial in the reverse order, according to (35), to obtain an allpass filter [34], [146], [71]. As can be seen in (31), the group delay of an allpass filter is twice that of its all-pole counterpart. Thus, Thiran's solution may easily be applied to the design of allpass filters by making the substitution $d' = d/2$. The solution for the coefficients of a *maximally-flat* (MF) *allpass filter* can be written in the form [34]

$$a_k = (-1)^k \binom{N}{k} \prod_{n=0}^{N} \frac{d+n}{d+k+n} \quad \text{for } k = 0, 1, 2, ..., N. \tag{35}$$

Note that $a_0 = 1$, and thus the coefficient vector need not be scaled. This closed-form solution is called the *Thiran allpass filter*. It approximates a group delay of $N + d$ samples, when the filter order is N and the fractional delay parameter is d. Since this parameterization may be rather impractical, we substitute $d = D - N$ into (35) so that the parameter D refers to the actual delay rather than the offset from N samples [174]. The coefficient formulas for the Thiran allpass filters of orders $N = 1$, 2, and 3 are presented in Table 2. The closed-form design of the first-order allpass filter was derived by using Taylor series expansion in papers by Jaffe and Smith [54], and Smith and Friedlander [153].

Table 2. Coefficients of the Thiran allpass filter for $N = 1$, 2, and 3.

	a_1	a_2	a_3
$N = 1$	$-\frac{D-1}{D+1}$		
$N = 2$	$-2\frac{D-2}{D+1}$	$\frac{(D-1)(D-2)}{(D+1)(D+2)}$	
$N = 3$	$-3\frac{D-3}{D+1}$	$3\frac{(D-2)(D-3)}{(D+1)(D+2)}$	$-\frac{(D-1)(D-2)(D-3)}{(D+1)(D+2)(D+3)}$

Thiran's proof of stability [166] implies that this allpass filter will be stable when $D > N - 1$. At $D = N - 1$, one of the poles (and one zero) of the Thiran allpass filter will be on the unit circle. When $D < N - 1$, one or more poles will be outside the unit circle and the filter will be unstable.

The suggested range of values for delay D to be approximated with the Thiran allpass filter is $[N - 0.5, N + 0.5)$. This choice is close to the optimum that minimizes the average frequency-response error for low-order allpass filters [174].

In the following, a design example is presented that can be compared with the FIR filter examples of Section 3. An MF allpass filter of order 4 is designed, to approximate a delay of 4.4 samples. The first 11 samples of its impulse response are displayed in Fig. 11(a). Note that the impulse response—although of infinite length in theory—decays rapidly, and it does not appear much different from those of FIR FD filters.

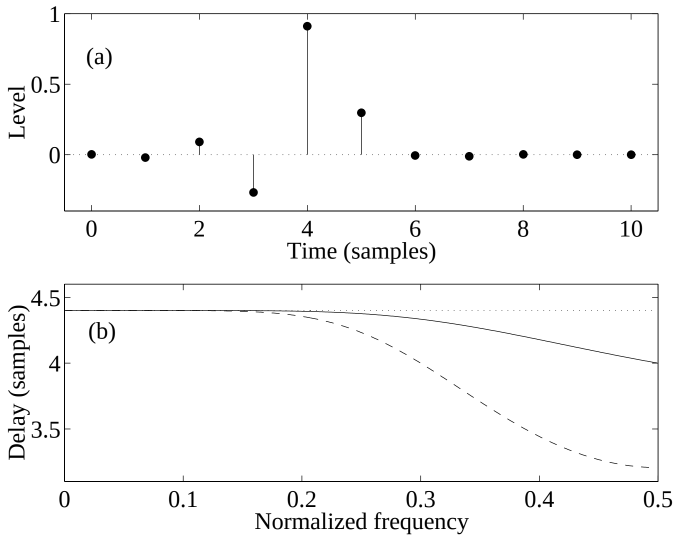

Fig. 11. (a) The beginning of the impulse response, and (b) the phase delay (solid line) and group delay (dashed line) of a Thiran allpass filter of order N = 4 approximating a delay of D = 4.4 samples.

Fig. 11(b) gives both the phase and the group delay of the allpass filter. They are practically equivalent at very low frequencies. Note that the phase delay behaves similarly to that of the Lagrange interpolator (see Fig. 8(d)).

Fig. 12(a) shows the frequency-response error of the allpass filter and that of a Lagrange interpolator of Fig. 8. Note that we have chosen the all-pass filter order so that very nearly the same error characteristics are obtained as in our previous Lagrange interpolator example. The allpass filter appears to be slightly better at normalized frequencies below 0.1 and slightly worse at higher frequencies, as illustrated by the error difference curve of Fig. 12(b). However, the allpass filter is more efficient in terms of computations, since only 4 multiplications and 7 additions are required per output sample; the Lagrange interpolator requires 8 multiplications and 7 additions. On the other hand, the delay caused by the allpass filter is one sampling interval more than that of the Lagrange interpolator.

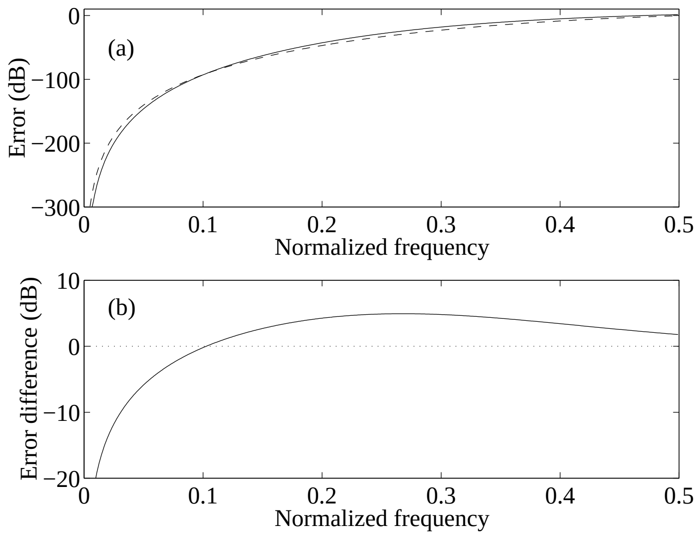

Fig. 12. (a) Comparison of the frequency-response error magnitude of the 4th-order allpass filter (solid line) of Fig. 11, and the 7th-order Lagrange interpolator (dashed line) of Fig. 8. (b) The difference of the error curves in (a).

4.2.3 Minimax Design of Allpass FD Filters

There exists a large variety of algorithms for equiripple or minimax allpass filter design in terms of phase, group delay, or phase delay [124], [132], [71]. New iterative techniques for equiripple phase error and phase delay error design of allpass fractional delay filters are described in [71]. Lang has developed a Remez-type algorithm that is guaranteed to converge to the optimal Chebyshev solution [74]. The method also alleviates the numerical problems that have been a major concern in designing high-order allpass filters.

4.3 Discussion

This section has considered the allpass filter approximation of fractional delay. A digital allpass filter is a good choice for this task since its magnitude response is exactly flat and the design can concentrate entirely on the delay characteristics. The Thiran allpass filter was discussed in detail since it is easy to design with closed-form formulas and is very accurate at low frequencies. One problem in the use of recursive FD filters is the *transient phenomenon* that occurs when the filter characteristics are changed during operation. This situation will be discussed in the following.

5. TIME-VARYING FRACTIONAL DELAY FILTERS

It is important to investigate the situation where the delay parameter of an FD filter is changed during operation, since in many real-time signal processing applications the desired delay is not constant but varies with time. In this section we study the implementation of a time-varying digital delay line using FIR and IIR filters.

5.1 Consequences of Changing Filter Coefficients

We now discuss how the change of the filter coefficients affects the output signal of the filter. A single change of the filter coefficients at time index $n = n_c$ is considered[†]. There may be two kinds of consequences at the same time [174], [180]:

1. The output signal may suffer from a *transient* that starts at $n = n_c$, and
2. the output signal may experience a *discontinuity* at $n = n_c$.

The *transient* at the filter output is observed if the state variables of the filter contain intermediate results related to the former coefficient set, or if the state variables are cleared. In the case of nonrecursive (FIR) filters this problem does not exist when the filter has been implemented in direct form (as shown in Fig. 3), but in the case of recursive (IIR) filters the occurrence of transients must be accounted for somehow. Fig. 13(a) gives an example: a sinusoidal signal is filtered with an allpass filter, and at time index 30 the coefficients of the allpass filter are changed. The transient is easily visible in the output signal of the allpass filter: the signal waveform deviates from the sinusoidal form for a few samples after time index 30.

There is also a *discontinuity* at the same time instant caused by the change of delay. A discontinuity means a sudden change in the corresponding continuous-time signal. The discontinuity of the output signal is simply caused by the fact that after the coefficients have been changed, the output signal will be a result of a different of filtering operation than

[†] Note that in this context the subscript ‘$_c$’ denotes ‘change’.

before the change. Usually this is, of course, a desired result, but the discontinuity may still be harmful in some applications. An example is given in Fig. 13(b): The transient in the signal of Fig. 13(a) has now been completely cancelled but the discontinuity remains. This case corresponds to time-varying IIR filtering with ideal transient cancellation or time-varying FIR filtering where transients do not occur. Note that there is only a small and abrupt single change in Fig. 13(b): the sample value at time index 30 is not part of the sampled sine wave displayed between indices 0 and 29; however, the waveform after time index 30 is seen to be a regular, periodically sampled sine wave. Thus, the signal is free from defects before and after the discontinuity.

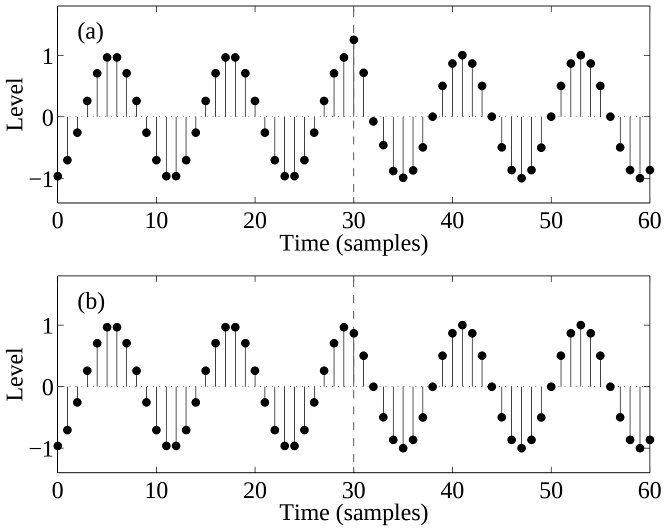

Fig. 13. Examples of (a) a transient and (b) a discontinuity in a signal waveform caused by time-varying fractional delay filtering. The vertical dashed line indicates the point of change.

Both the transient signal and the discontinuity depend on the input signal of the filter and the magnitude of coefficient change (and also on the time of change); a small change in the values of the coefficients causes a transient with smaller amplitude and a smaller discontinuity than a large change in coefficient values [180]. The severity of the transients and discontinuity can be decreased by making the change in filter parameters smaller, e.g., by allowing the filter a reasonably long transition time when the values of the coefficients are gradually changed (by interpolation) from the initial to the target values.

5.2 Implementing Variable Delay Using FIR Filters

We now examine the realization of a variable-length digital delay line using an FIR interpolating filter. Thereafter, design methods for time-varying FIR FD filters are reviewed.

5.2.1 Time-Varying Digital Delay Line

Fig. 14 shows the practical implementation of a given fractional delay using an FIR filter; part of the delay is implemented with a delay line of length M and the rest with an FIR filter that approximates the delay of D samples. The overall delay experienced by the signal $x(n)$ is thus $M + D$.

Let us consider the case when the delay parameter is a function of time, for example so that $D(n) = D_1$ for $n < n_c$ and $D(n) = D_2$ for $n \geq n_c$. If the next delay value D_2 does not fulfill the requirement $(N - 1)/2 \leq D_2 < (N + 1)/2$, the taps of the FIR FD filter must be moved to obtain the best approximation, i.e., the delay-line length M must be changed in Fig. 14.

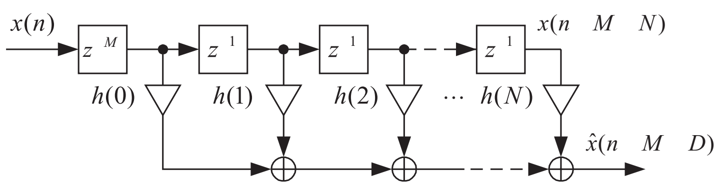

Fig. 14. A digital delay line implemented using FIR interpolation.

Care must be taken to correctly implement the delay change when $D_2 > D_1$ and the filter taps are moved; the delay line must be long enough so that the delayed sample values needed for filtering are immediately available. A good practical solution is to define a maximum delay D_{max} that is the assumed largest delay value (in samples) that will be needed in a particular application. The delay-line length is then set long enough for approximating a delay D_{max} using an Nth-order FIR FD filter. This implies that the delay-line length must be at least $M_{tot} = D_{max} + \text{floor}(N/2)$, where the extra floor($N$/2) unit delays are needed by the FIR filter when the largest delay D_{max} is approximated.

Note that a small delay $D_2 < (N - 1)/2$ cannot be approximated with good accuracy[†]. In this case, one should either use a filter of lower order N, or use the filter of original order outside its optimal range of delay parameter D. Examples with Lagrange interpolation have been presented in [174].

Thus, with careful design of the implementation, the transient signal can be completely eliminated when using an FIR interpolating filter. This is natural, since there is no feedback in the system in Fig. 14.

[†] This discussion is irrelevant for the case $N = 1$, e.g., linear interpolation, since for the first-order filter the smallest delay in the optimal range is zero.

5.2.2 Implementation of Time-Varying FIR FD Filters

There are two basic approaches to implementing a time-varying FIR filter, 1) a *table lookup method* and 2) *on-line computation of coefficients* [32], [31], [43]. In the table lookup method, the coefficients of all FD filters that will be needed are computed and stored in advance. When it is needed to change the delay parameter, the coefficients corresponding to the new delay value are retrieved from the memory. The main disadvantage of this approach is the need for a large memory that can be used for fast retrieval of data. For example, if the sampling interval is needed to divide into Q parts, and each fractional delay is approximated with an FIR filter of length L, the amount of coefficients to be stored is QL. A trivial way to reduce this amount by 50% is to store only the coefficients for the interval $0 < d < 0.5$, since the coefficients of FD filters for $0.5 < d < 1$ can be obtained from them by time reversal, as seen by comparing Fig. 5(b) and Fig. 5(d).

There is also a variation of the table lookup method, where some coefficients are stored in a table and the intermediate coefficient sets are obtained by polynomial interpolation [154], [122], [2]. Even the use of simple linear interpolation may reduce the amount of coefficient memory by more than 50% [154].

The main advantage of on-line computation of filter coefficients is the saving of memory. However, the computational complexity of coefficient update is usually larger than in the table lookup method. Laakso *et al.* [71] discussed the Fourier-transform method and a general Oetken's method for changing the delay of an FD filter. During the last decade, the Farrow structure has become the most popular method for implementing time-varying FD filters [32], [31], [173], [174], [43], [44], [188], [192], [189], [187], [190], [186]. This method is discussed in detail in the following sections.

5.2.3 Farrow Structure for Lagrange Interpolation

We now discuss a particularly efficient implementation structure for time-varying FIR FD filtering. Farrow [32] suggested that every filter coefficient of an FIR FD filter could be expressed as an Nth-order polynomial in the variable delay parameter D. This results in $N + 1$ FIR filters with constant coefficients. In this section we present an efficient structure for real-time implementation of Lagrange interpolation. This derivation has been adapted from [173] and [174]. The Farrow structure for Lagrange interpolation has been proposed first by Erup *et al.* [31].

Lagrange interpolation is usually implemented using a direct-form FIR filter. An alternative structure is obtained by approximating the continuous-time function $x_c(t)$ by a polynomial in *D*, which is the interpolation interval or fractional delay. The interpolants, i.e., the new samples, are now represented by the following function:

$$y(n) = \hat{x}_c(n - D) = \sum_{k=0}^{N} c(k) D^k \,, \tag{36}$$

that takes on the value *x*(*n*) when *D* = *n*. The above approach to Lagrange interpolation is seen to be related to Farrow’s idea of having *N* + 1 constant filters.

The alternative implementation for Lagrange interpolation is obtained by formulating the polynomial interpolation problem in the *z*-domain as *Y*(*z*) = *H*(*z*)*X*(*z*) where *X*(*z*) and *Y*(*z*) are the *z*-transforms of the input and output signal, *x*(*n*) and *y*(*n*), respectively, and the transfer function *H*(*z*) is now expressed as a polynomial in *D* (instead of z^{-1}).

$$H(z) = \sum_{k=0}^{N} C_k(z) D^k \,. \tag{37}$$

The familiar requirement that the output sample should be one of the input samples for integer *D* may be written in the *z*-domain as $Y(z) = z^{-D}X(z)$ for *D* = 0, 1, 2, ..., *N*. Together with (36) and (37) this leads to the following *N* + 1 conditions

$$\sum_{k=0}^{N} C_k(z)D^k = z^{-D} \text{ for } D = 0, 1, 2, ..., N. \tag{38}$$

This may be expressed in matrix form **Uc** = **z** where the matrix **U** has the Vandermonde structure and thus it has an inverse matrix $\mathbf{U}^{-1}$. The solution of (38) can thus be written as $\mathbf{c} = \mathbf{U}^{-1}\mathbf{z}$.

The transfer function $C_0(z)$ is equal to 1 regardless of the order of the interpolator. The other transfer functions $C_n(z)$ are Nth-order polynomials in z^{-1}, i.e., they are Nth-order FIR filters. This implementation technique is called the *Farrow structure of Lagrange interpolation*. A remarkable feature of this form is that the transfer functions $C_n(z)$ are *fixed* for a given order N. The interpolator is directly controlled by the fractional delay D, i.e., no computationally intensive coefficient update is needed when D is changed.

The Farrow structure is most efficiently implemented using *Horner's rule* (see, e.g., [51], p. 28), i.e.,

$$\sum_{k=0}^{N} C_k(z)D^k = C_0(z) + [C_1(z) + \ldots + [C_{N-1}(z) + C_N(z)D\overbrace{]D \ldots]}^{N}D\,. \tag{39}$$

With this method, N multiplications by D are needed. A general Nth-order Lagrange interpolator that employs the suggested approach is shown in Fig. 15. Since there is no need for the updating of coefficients, this structure is particularly well suited to applications where the fractional delay D is changed often, even after every sample interval.

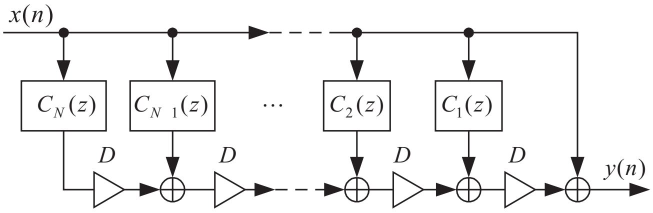

Fig. 15. The Farrow structure of Lagrange interpolation implemented using Horner's rule.

Fig. 16 shows the Farrow structure for first-order Lagrange interpolation. In this case the subfilters are $C_0(z) = 1$ and $C_1(z) = z^{-1} - 1$. The overall transfer function of the filter is $H(z) = C_0(z) + DC_1(z) = 1 + D(z^{-1} - 1)$.

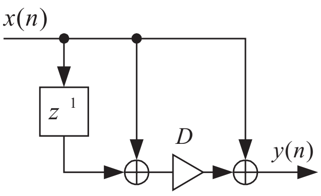

Fig. 16. The Farrow structure for linear interpolation.

If the delay is constant or updated infrequently, it may be more efficient to implement Lagrange interpolation using the standard FIR filter structure. Namely, with the FIR filter structure, $N + 1$ multiplications and N additions are needed. In Farrow's structure, there are N pieces of Nth-order FIR filters which results in $N(N + 1)$ multiplications and N^2 additions. There are also N multiplications by D and N additions. Altogether this means $N^2 + 2N$ multiplications and $N^2 + N$ additions per output sample, which is more than in the case of the standard FIR filter realization.

5.2.4 Design Methods for the Farrow Structure

In general, the order of the polynomials $C_n(z)$ in (37) and that of the time-varying FIR filter need not be the same in the Farrow structure [32], although this is the case when Lagrange interpolation is used. It is also possible to use optimization methods to design the polynomials that represent the coefficients of a time-varying FIR filter. This results in an FD filter whose delay characteristics can be controlled with the delay parameter D. Farrow gave an example where a seventh-order (8-tap) FIR FD filter was implemented using optimized polynomials of order three [32]. Laakso *et al.* [71] also proposed a general approach, where a set of polyphase filters is first designed, and a polynomial is thereafter fitted through each coefficient in the least squares sense, for example. This method is very general, since the number and length of polyphase filters, and the order of the polynomials can be chosen independently of each other.

Tarczynski *et al.* have suggested a technique where the polynomials $C_n(z)$ are designed by minimizing the weighted least-squares error criterion [162]. In the example given by Tarczynski *et al.*, the order of polynomials was 7 while the order of the FIR filter was 67, and the weighting function was a step-like function divided into five frequency bands [162]. The approximation error of the designed filter was kept less than –100 dB for $0 < d \leq 1$ at frequencies below $0.45F_s$.

Vesma and Saramäki [187], [188], [186] have presented a modified Farrow structure where the polynomials are functions of $2D - 1$ instead of D. The advantage is that the coefficients of the polynomial $C_n(z)$ will be symmetric or anti-symmetric, which may lead to computational savings in the implementation of the time-varying structure, and makes the optimization of the filter coefficients easier. Vesma and Saramäki have developed an optimization procedure for designing the polynomials so that the time-varying FIR FD filter will have equiripple properties [189], [190], [186].

Recently, harris stated that the subfilters of the Farrow structure are derivative filters [43]. The first subfilter $C_0(z)$ has ideally a flat magnitude response, $C_1(z)$ is a first-order differentiator, $C_2(z)$ is a second-order differentiator, and so on [43]. This knowledge may be used to facilitate the design of optimal subfilters for the Farrow structure. Interestingly, Sudhakar *et al.* have earlier tackled the design of interpolation filters based on FIR differentiators of different degrees [158].

5.2.5 Discussion

The use of a time-varying FIR FD filter (as opposed to a recursive one) is favorable in the sense that *no disturbing transients* will be generated when the delay D is changed. This is true, of course, only when the FIR filter has been implemented so that the input samples that are needed for computing the output of the filter after the change in the delay value are available (for example, a transpose form FIR filter causes transients if the filter coefficients are changed at the same time). Anyhow, a discontinuity is observed in the output signal of the filter, but it can be made small by making the coefficient changes small. The Farrow structure, or its modification proposed by Vesma and Saramäki [187], [188], [186], is an excellent candidate for realization of time-varying FIR filters.

5.3 Time-Varying Recursive FD Filters

An abrupt change in the coefficient values of an IIR filter gives rise to disturbances in the future values of the internal state variables and transients in the output values of the filter. These disturbances depend on the filter’s impulse response so that they are in principle infinitely long. In the case of stable filters, however, the disturbances decay exponentially and can be treated as having a finite length. In practice, these transients are often serious enough to cause trouble, such as clicks in audio applications, and they are typically the most serious problem in the implementation of tunable or time-varying recursive filters.

Despite the importance of the problem, there exists only a few research reports on strategies for elimination of transients in time-varying recursive filters (for references, see [174] and [180]). Zetterberg and Zhang presented the most general of these approaches by means of a state-space formulation of the recursive filter [200]. Their main result was that, assuming a stationary input signal, every change in the filter coefficients should be accompanied by an appropriate change in the internal state variables. This guarantees that the filter switches directly from one state to another without any transient response. The Zetterberg–Zhang (ZZ) method can *completely eliminate the transients* but it does require that all the past input samples are known. For this reason, the ZZ algorithm is impractical as such but has provided a good starting point for a more efficient approximate algorithm described in [181], [174], and [180].

The transient-cancellation algorithm uses two filters in parallel, as shown in Fig. 17: one for filtering the signal with current filter coefficients, and another with the next filter coefficients; the output signal of the secondary filter is not used at this time. Two filters must be executed in parallel only for a finite number of sampling intervals, then the outputs of the filters can be switched and the old filter can be stopped. This is motivated by the fact that the impulse response of a stable recursive filter decays exponentially and can thus be regarded as finite-length. The knowledge of the effective length of the impulse response from the input to the state vector helps to estimate how many past input samples need to be taken into account in updating the state vector of the filter to be used next. Thus, the advance time may then be set equal to $N_a = N_P + N$ where N_P is the effective length of the impulse response and N is the order of the filter [180]. This choice of N_a ensures that the updated state vector suffers sufficiently little from the truncation of the input signal, according to the same criterion that was used to determine N_P. In practice, it is desirable to choose N_a to be the smallest integer that yields sufficient suppression, since this minimizes the implementation costs of the transient cancellation algorithm. The effective length of an infinite-length impulse response can be determined using an energy-based method [69].

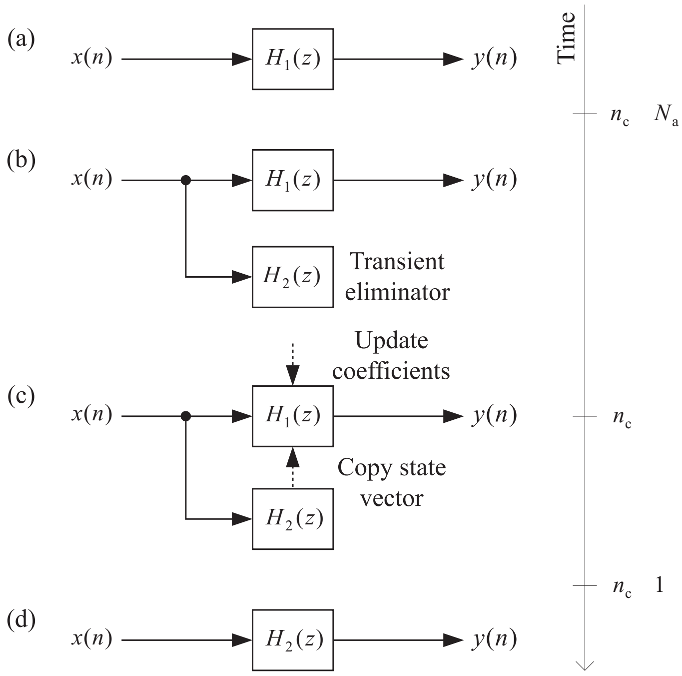

Fig. 17. Different phases of the transient suppression scheme for a single change of filter coefficients (adopted from [180]).

For a single coefficient change, the algorithm requires that two filters run in parallel for N_a sample intervals, as illustrated in Fig. 17. Thus, when multiple changes are required and it is fast enough to update filter coefficients at every N_ath sample interval, there is no need to run more than two filters in parallel at any time. The main advantages of this technique is that now the computation of the transient cancellation vector only takes finite time and need not be updated all the time in parallel with the filtering operation. Also the accuracy of transient cancellation can be controlled with parameter N_a: the larger the value of N_a, the more transient suppression is achieved.

5.3.1 Examples on Transient Suppression

We present an example that illustrates the transient suppression method. We filter a low-frequency sine wave (0.0454 times the sampling frequency F_s) with a second-order allpass filter (direct-form II) that approximates a constant group delay. Initially, the filter coefficients are $a_1 = 0$ and $a_2 = 0$ (corresponding to a constant delay of 2 samples) and at time index 30 they are changed to values $a_1 = 0.4$ and $a_2 = –0.028571$, which gives a group delay of 1.5 samples at low frequencies. We present the output and transient signals of the filter in two cases: without transient cancellation and when the cancellation method is used with parameter value $N_a = 4$. These output signals are compared with the “ideal” output signal—see the open circles in Fig. 18(a) and Fig. 19(a)—which has been computed using the output-switching method (by running two filters in parallel and changing the output at time $n = 30$). The transient signal shown in the lower part of the figures in both cases is the difference of the output signals of the time-varying and ideal filter. Obviously, in Fig. 19 ($N_a = 4$) the maximum amplitude of the transient has been suppressed with respect to Fig. 18. More suppression can be achieved by using a larger value for N_a.

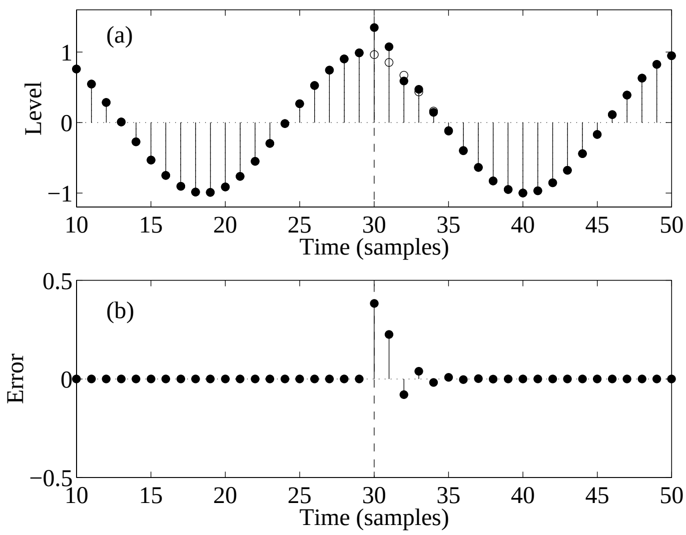

Fig. 18. (a) Output signal and (b) transient of a second-order allpass filter (DF II structure) when the coefficients are changed at time index 30 without transient elimination.

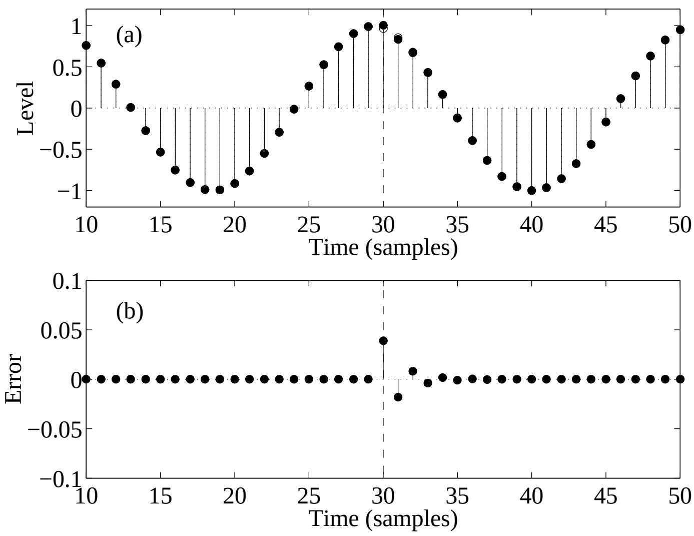

Fig. 19. (a) Output signal and (b) transient of a second-order allpass filter (DF II structure) when the coefficients are changed at time index 30 with the transient suppresser with $N_a = 4$.

5.4 Conclusions

We have discussed time-varying fractional delay filters. The FIR filters do not suffer from transients when their coefficient values or locations of filter taps are changed. The output signal of an FIR filter will, however, be discontinuous at the time of the change. A strategy for efficient implementation of time-varying FIR FD filters, known as the Farrow structure, was presented. Techniques for designing time-varying FIR FD filters were also surveyed.

Recursive filters, such as allpass FD filters, suffer from two problems: transients and discontinuities in the output signal. There is a method that can suppress the transient as much as needed by increasing the advance time parameter. The required amount of computations is directly proportional to this parameter.

6. FRACTIONAL DELAY FILTERS FOR NONUNIFORMLY SAMPLED SIGNALS

In this section, we consider fractional delaying of a nonuniformly sampled signal, where the sampling interval changes from sample to sample. The reconstruction of the original continuous-time signal or resampling onto a different (usually uniform) grid are relevant practical problems where interpolation techniques similar to uniform FD filters can be used. We discuss practical approaches to interpolation of nonuniformly sampled signals by a *generalized fractional delay filter* which takes as its input the signal samples with timing information, and produces estimates for the signal values at the desired sampling instants.

The uniform FD filtering problem discussed in previous sections can be viewed as an application of the reconstruction formula (9) of bandlimited uniformly sampled signals. If we retain the bandlimited signal but assume nonuniform samples, the reconstruction problem can still be solved based on the sinc series. However, the solution becomes more involved both in terms of computational complexity and numerical problems [84], [99]. Several approximate methods have been proposed which often include iterative techniques [85].

One approach to circumvent the numerical problems of the ideal bandlimited sinc reconstruction is to use *polynomial interpolation*. In the most straightforward case it reduces to Lagrange interpolation, i.e., fitting of a polynomial of order N through a set of $L = N + 1$ sample values. The coefficients of the general Lagrange interpolator filter for a set of N uniformly sampled input signal values t_n (where $n = 0, 1, 2, ..., N$) are obtained as

$$h(t - t_n) = \prod_{\substack{k=0 \\ k \neq n}}^{N} \frac{t - t_k}{t_n - t_k} \quad \text{for } n = 0, 1, 2, \ldots, N, \tag{40}$$

where t is the value of the desired output time. The interpolated output signal is obtained via nonuniform convolution as

$$x(t) = \sum_{n=0}^{N} h(t - t_n) x(t_n) \, . \tag{41}$$

Note that now the filter coefficients are time-varying—unlike those of the uniform Lagrange interpolation (see Section 3.4)—and generally depend on all the sampling instants on the particular set of L samples. If an additional fractional delay t_D is desired, the time variable t should be replaced with (t –

t_D). The use of Lagrange interpolation for reconstruction of nonuniformly sampled signals has been studied by Murphy *et al.* [102].

The problem with Lagrange interpolation (as with the ideal sinc reconstruction as well) is that it makes an exact match with the signal at the sample instants. This is often unnecessary, and may even be harmful in practice when signal samples to be processed contain additive noise. In such a case, we would instead like to suppress the noise and reconstruct only the desired part of the signal.

Knowing what the 'desired part of the signal' is, requires *a priori* knowledge. Often the desired part of the signal is slowly varying, which can be modeled by assuming the signal to be bandlimited up to a certain frequency. Alternatively, the signal can be assumed to be a polynomial of a certain degree P (less than $N-1$). This is a generalization of Lagrange interpolation which assumes that the sequence of N samples can be modeled by a polynomial of order $N-1$ and implements exact curve fitting at the desired sample values. This kind of polynomial filtering, besides being less intensive computationally than Lagrange interpolation, enables simultaneous reconstruction of the nonuniform signal and suppression of wideband noise [68].

The polynomial model for the signal can be expressed in the form

$$p(t) = \sum_{k=0}^{P} c_k (t - t_0)^k , \tag{42}$$

where P is the order of the polynomial, c_k are the polynomial coefficients, and t_0 is the time instant of the first sample. Minimizing the mean squared error (MSE) of the polynomial model and signal samples results in a set of normal equations for the polynomial coefficients the solution of which can be expressed in matrix form as

$$\mathbf{c}_{\mathrm{opt}} = \left[\mathbf{U}^{\mathrm{T}}\mathbf{U}\right]^{-1} \mathbf{U}^{\mathrm{T}}\mathbf{x} , \tag{43}$$

where $\mathbf{c}_{\mathrm{opt}}$ is the vector of the P + 1 polynomial coefficients, $\mathbf{U}$ is a matrix depending on the input sampling instants, and $\mathbf{x}$ is the vector of N input signal values (for details, see [68]). The new sample values are obtained by evaluating the polynomial (42) at the desired sampling instant. Similarly to uniformly sampled signals, the best results are obtained when the interpolation is carried out for samples near the center of the time window of the input samples.

Let us consider an example. We assume a nonuniformly sampled signal generated by a *jittered* sampling process, i.e. a nominally uniform sampling rate $F_s = 1/T$ which is distorted by random timing errors that have a nonuniform distribution with $\Delta = T/4$. The signal consists of a sum of low-frequency sinusoids at frequencies $f_1 = 0.070F_s$ and $f_2 = 0.123F_s$ with total power equal to unity. In addition, the signal is corrupted with additive uncorrelated Gaussian noise of variance 0.15, i.e., the signal-to-noise ratio (SNR) is 8.2 dB.

Fig. 20(a) shows the original sinusoids without noise (uniformly sampled) and the nonuniformly sampled signal corrupted with noise. Only the noisy nonuniformly sampled signal is available in a practical situation, and it is used to produce the uniformly sampled reconstructions shown in Fig. 20(b). Note that, using the polynomial interpolation technique described above, the signal could have been also delayed by a fraction of the nominal sampling interval, or oversampled on a more dense grid, but here we focus on the simple case. Seven-tap (L = 7) polynomial filters were used. We first applied Lagrange interpolation (i.e., polynomial order $P = L - 1 = 6$) and then the polynomial filter with P = 3. It is seen that the Lagrange interpolation produces a signal that is quite a faithful reconstruction of the noisy one, whereas the polynomial filtering result is more similar to the noiseless original signal. The corresponding MSE values are 0.1486 and 0.0480 (corresponding to noise reduction by 0.041 dB and 4.9 dB, respectively), demonstrating that the polynomial filter is able to suppress the noise by almost 5 dB and reconstruct the desired part of the signal accurately at the same time.

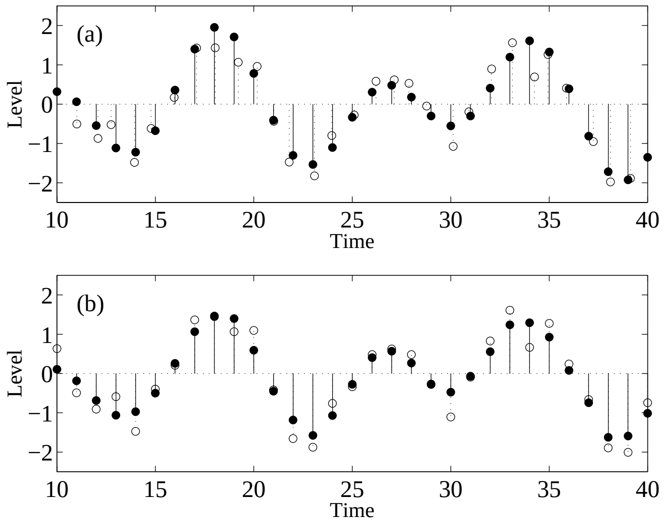

Fig. 20. (a) Original uniformly sampled signal (l) and its noisy, nonuniformly sampled version obtained by jittered random sampling (m). (b) Comparison of seven-tap polynomial reconstruction using 6th-order Lagrange interpolation (m) and 3rd-order polynomial interpolation (l).

7. APPLICATIONS OF FRACTIONAL DELAY FILTERS

Fractional delay filters can be utilized in many areas of digital signal processing. In the following, we discuss some technical fields where FD filters are useful or necessary, including sampling-rate conversion, timing adjustment of digital modems, music synthesis, virtual audio reality, array beamforming, and digital filter design. Other examples where fractional delay filters have turned out to be helpful are time delay estimation [153], [27], [28], [29], speech coding [82], [83], [65], [89], [3], [15], [88], stabilization of feedback systems [160], timing correction in multichannel data acquisition systems [6], and image resampling [52], [62], [26]. The key ideas in the usage of FD filters in several applications were also discussed in the review article by Laakso *et al*. [71].

7.1 Sampling-Rate Conversion

Changing the sampling rate is one of the standard problems in digital signal processing. A well-known example is the conversion between the different sampling frequencies used in digital audio, which include 44.1 kHz (used in the CD-quality audio), 48 kHz (used by the DAT recorders), 32 kHz, 22.05 kHz, and 11.025 kHz among others (see, e.g., [116]).

The sampling-rate conversion is straightforward if the ratio of the input and output sampling rates is an integer or a ratio of small integers [19], [172]. Efficient polyphase filter structures are known for the implementation. As discussed in Section 3.1, each branch of the polyphase filter can be interpreted as a lowpass FD filter approximating an integer number of fractions of the input sample interval.

The sampling-rate conversion for incommensurate ratios is more complicated. In practice, the situation is often made even harder by the fact that the sampling-rate ratios are not only irrational but also *time-varying*, which is caused, for instance, by variations in clock frequencies due to temperature, aging, or external disturbances.

A straightforward solution to the incommensurate sampling-rate conversion problem can be obtained via an extension of the polyphase implementation. Any ratio can, of course, be approximated to a desired precision by a ratio of large integers. Hence, one just needs sufficiently many polyphase branches to have a dense grid of preset polyphase branches from which to choose. In other words, a large bank of predesigned FD filters is needed. In addition, in order to avoid *folding*, care must be taken in the case of downsampling that the signal to be resampled is bandlimited below the final Nyquist frequency: then the polyphase filters must be fractional delay lowpass filters.

From the polyphase approach there is conceptually only a short step to using time-varying FD filters; instead of storing a large number of filter coefficients one can design them on-line, e.g. by employing the above presented Farrow structure.

The incommensurate sampling-rate conversion problem can be illustrated by Fig. 21 where the sampling time instants of both the input and output sampling rates are shown. It is obvious that every output sample can be obtained by applying an FD filter approximating an appropriate delay. The essential tasks are thus to implement a *control unit* that computes the required delay values $D(n)$ and a computationally efficient FD lowpass filter whose coefficients change for every output sample. A block diagram is shown in Fig. 22.

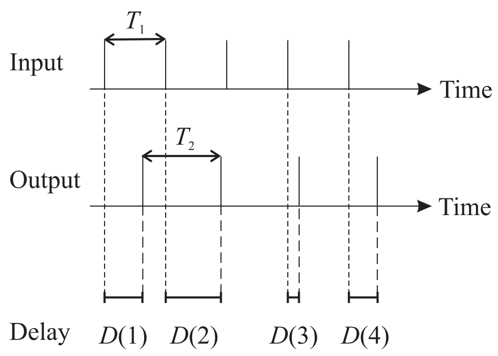

Fig. 21. Illustration of the sampling grid of two signals sampled at rates f_{s1} and f_{s2}, respectively, whose sampling intervals are T_1 and T_2, and the fractional delays $D(n)$, between the contiguous samples of these signals.

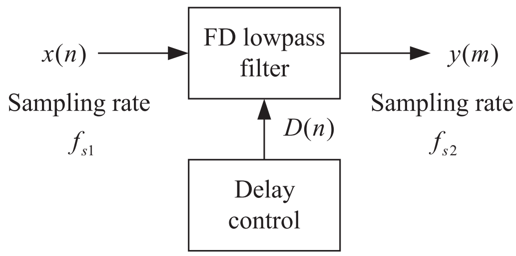

Fig. 22. Block diagram of an incommensurate-ratio sampling-rate converter.

Previous work on incommensurate sampling-rate conversion includes papers by Smith and Gossett [154], Ramstad [122], [123], de Carvalho and Hanson [14], Cucchi *et al.* [20], Park *et al.* [111], Adams and Kwan [2], Tarczynski *et al.* [163], Zölzer and Boltze [204], [205], Laakso *et al.* [70], Saramäki and Ritoniemi [134], and Murphy *et al.* [103].

Arbitrary changes of sampling rate are also required in wavetable music synthesis, where a sampled waveform is stored in a buffer and during playback the buffer is read repeatedly to synthesize a periodic tone [10], [86]. The pitch of the note during playback is controlled by the increment of the read pointer to the buffer. For example, if a tone one octave higher than that stored in the buffer is required, the increment size is 2 samples. An increment size of 4 would result in a tone that is 2 octaves higher than the one stored, and so on. For lower tones and for other intervals than simple integer ratios, a technique called fractional addressing must be used [46]; then the increment size is not an integer. In such cases, simply selecting the nearest sample in the table causes errors in the synthetic waveform which may not be acceptable [98], [21], [86]. Fractional delay filtering techniques can be used to approximate the signal value between the stored samples during playback [98], [193], [21].

Another example of a related audio signal processing application, where the signal must be resampled but the sampling rate does not necessarily change, is restoration of old recordings, particularly the elimination of wow in gramophone disc or magnetic tape recordings [40], [41].

7.2 Synchronization in Digital Receivers

In digital data transmission, the transmitter transmits data symbols using analog waveforms which carry one or more bits of information each. The main task of the receiver is to detect these symbols as reliably as possible. To this end, the receiver must be synchronized to the symbols of the incoming data signal. Even though the receiver usually knows the nominal symbol rate of the transmitter, the analog oscillators that are used to generate the frequency in practice have a limited precision, and the frequency tends to vary with temperature and aging. Furthermore, in mobile communications frequency shifts due to the Doppler effects must also be accounted for. Hence, in practice the synchronization requires constant monitoring and adjustment.

The symbol synchronization has traditionally been implemented by using an analog feedback or feedforward control loop to adjust the phase of a local clock at the receiver so that the sampling frequency and the sampling instants are adapted to the incoming data signal (see, e.g., [93]). An example of this kind of a receiver is illustrated in Fig. 23.

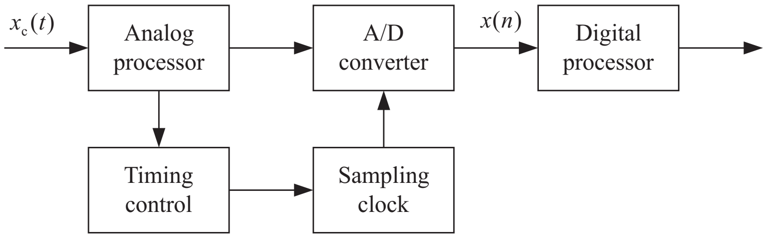

Fig. 23. Analog synchronization via controlling the sampling time (adapted from [36]).

Since the postprocessing of the sampled signal (matched filtering and data detection) is usually performed digitally anyway, it is also advantageous to implement the synchronization using digital techniques. A basic digital solution is outlined in Fig. 24. The local oscillator is now independent of the received signal and often operates at a higher rate than the nominal symbol rate (e.g. two or four times oversampled). The raw samples are now postprocessed by an appropriate FD filter. Especially in the case of oversampling, simple low-order FD filters (such as a 4-tap Lagrange interpolator) are usually sufficient [70]. For on-line control of the delay value, the Farrow structure has been found practical.

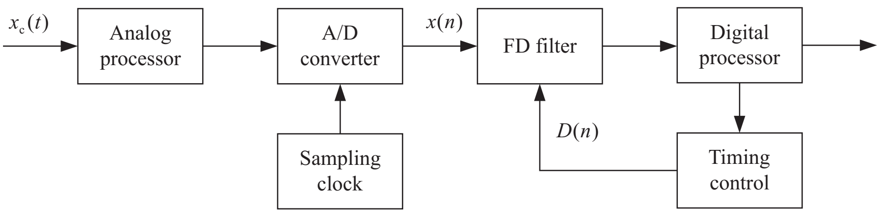

Fig. 24. Digital synchronization via an FD filter (adapted from [36]).

For excellent tutorials to synchronization in digital receivers, see [36], [31], [90] and [94]. Related research papers include [32], [5], [45], [39], [185], [70], [203], [109], [168], [201], [202], [194], [188], [197], [192], [187], [73], [35], and [61]. For a more efficient implementation and lower overall delay, filter structures which integrate the FD filter and other filtering functions, e.g., matched filtering and interference suppression, have been investigated in [127], [129], [128], [66], [125], [198], [199], and [191].

7.3 Music Synthesis Using Digital Waveguides

Fractional delay filters are essential in music synthesis based on digital waveguide modeling [149], [54], [159], [59], [150], [175], [151], [152]. A one-dimensional acoustic resonator, such as a vibrating string, or the tube of a wind instrument, can be modeled with a bi-directional delay line. In the computational model, the two delay lines can usually be combined into one. Generally, a fractional delay filter is required for fine-tuning the delay, which determines the pitch of the synthetic sound.

Fig. 25 presents a plucked string synthesis model. The loop filter is a low-order digital filter (e.g., a first-order recursive filter) that simulates the attenuation experienced by the string vibration. The propagation delays of transversal waves along the string have been combined into a single delay line. The FD filter adjusts the overall loop delay so that the pitch of the synthetic tone is correct. The input signal of the synthesis model can be a short burst of noise, but for more sophisticated synthesis the excitation signal can be extracted from a recorded plucked string tone using an inverse filtering technique [175]. The fractional delay filter used in plucked string models is usually either a first-order allpass filter [54] or a low-order Lagrange interpolation filter [59], [175].

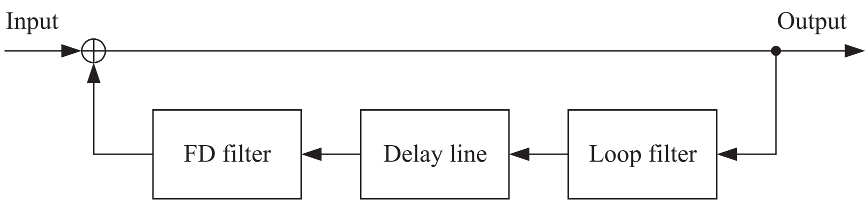

Fig. 25. Block diagram of a plucked string synthesis model (after [54]).

If it is desired to vary the pitch of the synthetic tone to produce for example glissando or vibrato effects and a recursive FD filter, i.e., an allpass filter, is used, transient problems will occur. This will appear as so-called zipper noise in the output audio signal. The transient cancellation method addressed previously in this chapter has turned out to be helpful reducing this effect [181]. Van Duyne and others have proposed a technique which combines the transient elimination method and cross-fading [184].

Fractional delay filter techniques have also been useful in adjusting the length of a vocal-tract model based on the transmission-line method, which is very similar to the digital waveguide approach [156], [72]. The locations of scattering junctions in a vocal tract model can also be fine-adjusted using interpolation techniques [178], [177], [176], [174]. This approach is called fractional delay waveguide modeling. The interpolated scattering junctions can be generalized for three waveguide branches, which is useful for implementing accurately the location of finger holes in woodwind instrument synthesis models, e.g., for the flute [179], [174], [17], [141].

Recently, a new concept in physics-based music synthesis was introduced, the passive nonlinearity [115]. It refers to a nonlinear phenomenon which enables energy dissipation such as in natural linear systems. This kind of phenomenon is very pronounced in metallic percussion instruments, such as gongs, but has also been discovered in string and wind instruments.

In synthesis models, passive nonlinearities can be realized using time-varying fractional delay filters with signal-dependent coefficients. Synthesis models based on this approach have been proposed that incorporate a longitudinally yielding (springy) end-point of strings [115], modulation of tension of a vibrating string [183], [167], and nonlinear sound propagation caused by high pressure in brass instruments [97], [165]. The time-varying FD filter may be based on a first-order allpass filter [115] or an FIR filter [182].

7.4 Other Musical Applications of Fractional Delays

Fractional delays are useful in digital implementation of traditional electronic musical effects, such as chorus and flanger, which are based on summing the original musical signal and its delayed version [22], [33]. A new way of implementing a time-varying delay line for musical applications using a circular buffer has been proposed by Rocchesso [126].

Also, a surround sound system can be constructed using two loudspeakers by filtering their input signals in an appropriate way [56], [57], [38]. The signal processing part of the system is composed of digital filters that approximate head-related transfer functions of the listener, and fractional delay filters that accurately model acoustic propagation delays (see [38], pp. 65–73).

Two advanced applications where FD filters are needed in the simulation of acoustic propagation delay are addressed in the following.

7.4.1 Doppler Effect in Virtual Audio Environments

A special feature of acoustic signals is their relatively slow propagation speed in air (ca. 340 m/s). For this reason sound is considerably delayed when it propagates from the sound source to the listener. This phenomenon needs to be modeled using delay lines and fractional delay filters in virtual reality applications that try to provide a realistic, immersive sound field [136]. When the sound source and the listener are allowed to move fast, the Doppler effect is heard. The Doppler shift refers to the rise or fall of the pitch respectively when the distance between the listener and source is increasing or decreasing fast enough. More generally, the Doppler effect means a scaling of the spectrum of a sound signal observed when the distance to the source varies over time. Fig. 26 illustrates how the propagation delay of sound varies when a car passes a listener at a speed of 100 kilometers per hour. The shortest distance between them is 4 meters which occurs at time t = 0 s. The time-varying delay given in Fig. 26 should be implemented with the help of fractional delay filters to bring about the Doppler shift. Strauss has discussed the use of interpolation filters in simulation of the Doppler effect in an auditory virtual reality application [155].

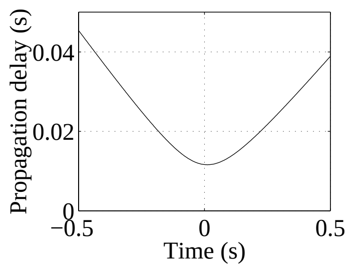

Fig. 26. Variation of propagation delay of sound from a car when it passes a listener with the speed of 100 kilometers per hour (after [155]).

7.4.2 Simulation of Wave Propagation in Multiple Dimensions

A sophisticated application of fractional delay filtering is a new method for simulation of multidimensional wave propagation using a finite difference mesh [137], [138], [135], [139], [140]. The basic model, the digital waveguide mesh, is based on a multidimensional extension of digital waveguides discussed in the previous section. It is a computationally efficient way of discretizing a membrane or an acoustic space, but the problem is that sample updates only occur in a limited number of directions, e.g., only 4 in the two-dimensional rectangular mesh. This limitation appeared to be the cause for fluctuation in the wave propagation speed in different directions and at different frequencies. In other words, the simulated mesh suffers from direction-dependent dispersion. Savioja and Välimäki extended the waveguide mesh method by applying fractional delay filters in multiple dimensions to devise a sample update procedure that renders the mesh more homogeneous. Improved performance has been obtained in both two [140] and three dimensions [135]. The method is applicable to simulation of drums and computation of the impulse response of acoustic spaces, e.g., listening rooms or concert halls.

7.5 Interpolated Array Beamforming

A classical method of beamforming in antenna or transducer arrays is based on delaying and summing the input signals of the array elements. Pridham and Mucci introduced *digital interpolation beamforming* in the 1970s [118], [119]. Their approach was to use polyphase FIR filters to accurately implement the required time delays. A similar solution has been proposed also recently [143]. Interpolated beamforming is applicable to sonar [118], [119], antenna arrays [76], and microphone arrays [42], [101].

Fig. 27 shows the configuration for traditional delay and sum beamforming. For the array elements, we use the symbol of a microphone but they could also be hydrophones, antenna elements, or other detectors. Notice that we also do not show the AD converters and other electronics required, but simply assume that the input signal obtained from each element is a discrete sequence $x_k(n)$ for $k = 1, 2, ..., M$, where M is the number of array elements. We also assume for simplicity that the spacing of array elements is uniform. The time delays D_k should be selected so that the different channels are in the same phase for a plane-wave signal which is incident from a specific direction θ. The summing causes the signal components incident from other directions to sum destructively. Obviously, the time delays D_k used in delay and sum beamforming must be realized accurately to obtain good directivity properties. If integer-length delay lines are used, it is possible to obtain accurate directivity in M directions between 0 and π (or between $-\pi$ and 0). When fractional delay filters are used to implement delay lines, a continuum of directions between 0 and π is obtained. The quality of the beamformer then depends on the quality of the fractional delay filters used.

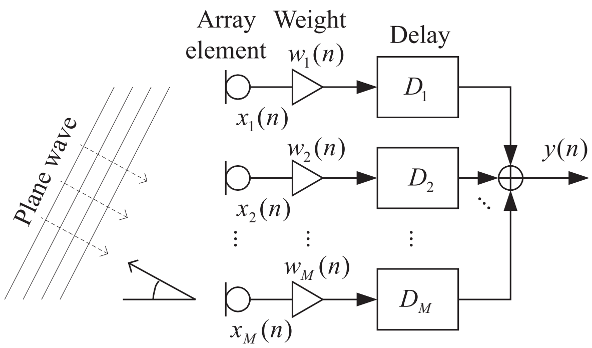

Fig. 27. Delay and sum beamforming using an array of sensors, such as microphones.

We present a design example where the array consists of 21 omnidirectional sensors whose distance has been set so that it corresponds to 1 sampling interval. The steering angle has been selected to be $\pi/5$, or 0.63 rad. With this choice all but one of the delays have a non-integer value. The first delay line can be neglected and replaced with unity gain, of course. The weighting coefficients for the signals of each sensor were taken from a Hamming window function. Fig. 28 shows with a solid line the beam pattern of the array evaluated at the normalized frequency 0.25 as a function of incident angle. The beam pattern has been normalized so that the maximum is at 0 dB. We also incorporated Lagrange interpolation filters into the delay lines, and repeated the beam pattern evaluation with two different orders, 2 and 10, shown with dashed and dash-dot lines in Fig. 27. Note that in this case the main lobe of the beam pattern is not much affected by fractional delay filters but the attenuation of interfering directions is improved generally about 2 to 3 dB. The difference between the performances of the two choices of Lagrange FD filter orders is small, because at the normalized frequency 0.25 both filters provide a good approximation.

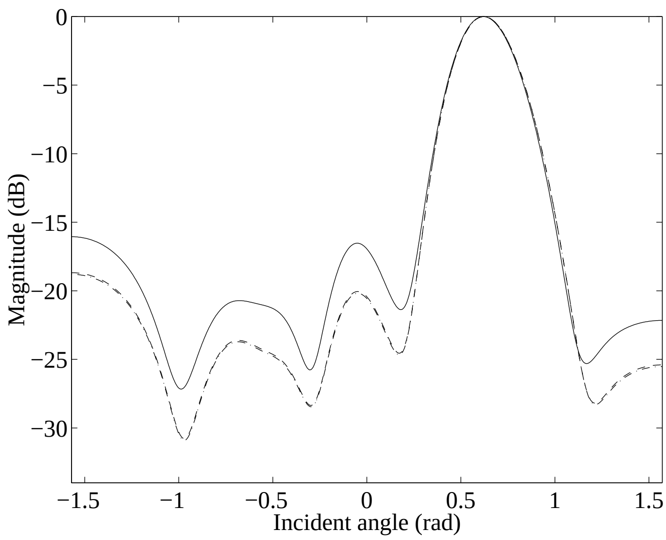

Fig. 28. Beam pattern of a 21-element array at the normalized frequency 0.25 with integer-length delay lines (solid line) and with second (dashed line) and tenth-order (dash-dot line) Lagrange fractional delay filters. The steering angle is 0.63 rad.

7.6 Design of Special Digital Filters

The fractional delay property can be combined with different digital filter characteristics. The control of delay of arbitrary FIR filters has been tackled by Adams [1] and Laakso *et al.* [71]. Hermanowicz and Rojewski [49] have derived closed-form equations for maximally-flat FIR differentiators with arbitrary fractional delay. Recently, Hermanowicz *et al.* have discussed also other digital filters with fractional delay, such as the Hilbert transformer, the half-band filter, and a complex-valued filter for computing the analytic signal [50]. Braileanu has discussed digital filters with explicit interpolated output [9].

Nazra has developed a design method for recursive fractional delay lowpass filters [104]. The method is based on using the Thiran fractional delay all-pole filter (discussed earlier in this chapter) as the recursive part of the filter, and then introducing zeros by designing an FIR compensator. The overall filter has lowpass characteristics with an equiripple stopband and adjustable fractional delay.

Selesnick generalized the maximally flat allpass filter design method into a filter that uses only some degrees of freedom for flatness constraints at the zero frequency and the rest to flatten the response at the Nyquist frequency [145]. A parallel sum of two such allpass filters can be used to implement lowpass (or highpass) filters with a few multiplications. The resulting structure is numerically robust against quantization effects.

Recently, the use of a polyphase FIR structure that implements fractional delays was found to be a fundamental issue in the design of a delayless sub-band adaptive filter [91], [25], [92].

Fractional delay filters can also facilitate the design of special comb filters, which we discuss in the following.

7.6.1 Fractional Delay Comb Filter

Pei and Tsang have used fractional delay filters to control the frequencies of notches in comb filters' transfer function [113]. The transfer function of the fractional delay comb filter is

$$H_{\text{fdc}}(z) = \frac{1 - H(z)}{1 - \rho^D H(z)}, \tag{44}$$

where $H(z)$ is the transfer function of a fractional delay filter approximating a delay D, and ρ is the radius of the poles used to control the flatness of the magnitude response between to the notch frequencies ($0 < \rho < 1$). Fractional delay comb filters can be used to cancel harmonic disturbances, such as power line interference in electrocardiogram (ECG) signals (see [113] for an example of this application).

We present an example on the use of the fractional delay comb filter. Let us assume that the sampling rate is 490 Hz. In Europe, the power line interference occurs at 50 Hz and its multiples. The four lowest normalized frequencies where the interference may occur are 0.1020, 0.2041, 0.3061, and 0.4082. If an integer-length delay line is used, the notches are closest to these frequencies for a comb filter order $N = 10$. The normalized frequencies of its transfer-function zeros are 0, 0.1, 0.2, 0.3, 0.4, and 0.5. The magnitude response of the comb filter with $\rho = 0.98$ is shown in Fig. 29(a) together with the dashed vertical lines that indicate the four interfering frequencies. The match between the notches and the interferences is quite poor giving an attenuation of 4.6 dB, 1.2 dB, and 0.2 dB for the three lowest frequencies, and an amplification of 0.24 dB for the fourth one. Note that the gain of the comb filter exceeds unity between the notches, the maximum gain being about 1.1 or 0.83 dB.

Instead of a delay of 10 samples—as in the comb filter of Fig. 29(a)—a delay of 1/0.1020 or 9.8 samples is required to exactly cancel a periodic disturbance of fundamental normalized frequency 0.1020. The fractional delay can be realized using one of the techniques discussed previously. We have chosen a Lagrange interpolation filter of order 4 in this example. The fractional delay parameter of the Lagrange interpolation filter must be set to

–0.2 samples, so that the effective delay of the comb filter is shortened to 9.8 samples, as desired.

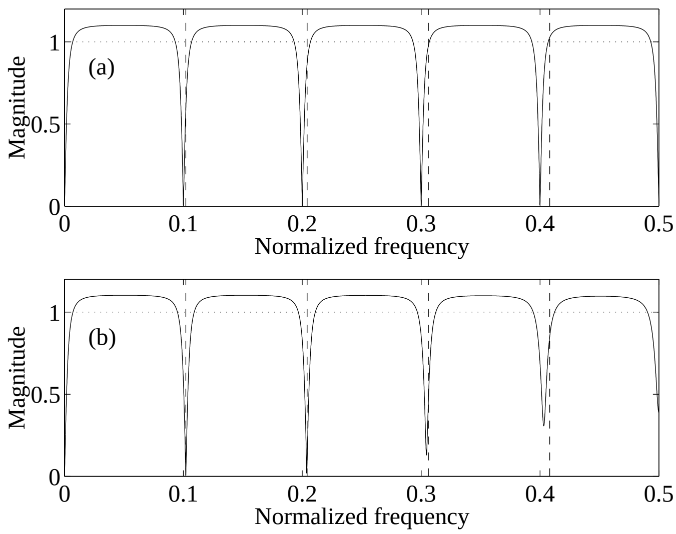

Fig. 29. Comparison of the magnitude responses of (a) a conventional comb filter (N = 10, ρ = 0.98) and (b) a fractional delay comb filter (ρ = 0.98, d = –0.2), which contains a Lagrange interpolator of order 4. The vertical dashed lines indicate the disturbing frequencies, which should be cancelled. The horizontal dotted line indicates the unity gain.

Fig. 29(b) shows the magnitude response of the fractional delay comb filter together with the interfering frequencies. It is seen that the match is excellent at the lowest two frequencies while at the third and fourth ones it is not perfect, but it is still better than in Fig. 29(a). Note also that the third and the fourth notch do not go to zero. This is caused by the lowpass character of the Lagrange interpolator. The attenuations at the four frequencies of interest are 49 dB, 20 dB, 6.6 dB, and 1.4 dB. The improvement over the conventional comb filter performance is dramatic at low frequencies, but at high frequencies the advantage is smaller. The result could be improved using a fractional delay design technique that optimizes the performance at the frequencies of interest, or alternatively, a general wideband FD filter design method could be used.

The behavior of the FD comb filter of this example could also be desirable if the interfering signal were also of lowpass character, i.e., having a strong fundamental component plus weaker harmonic components whose amplitude is inversely proportional to frequency; then it would be useful to have more attenuation at low frequencies where the strongest interfering harmonics appear.

The main advantage of the example design presented in Fig. 29(b) is that the remarkable improvement is achieved with a simple FD filter which can be adapted using only one parameter, the fractional delay d.

8. CONCLUSIONS

This chapter covered the theory and applications of fractional delay filters. Fractional delay filters are helpful in a wide range of signal processing applications where the temporal resolution of a fixed sampling interval is not enough. This is the case in problems involving detection, synthesis, or synchronization of signals, such as in time delay estimation, array beamforming, sampling-rate conversion, speech and music synthesis, time-domain simulation of acoustic systems, and synchronization of digital modems. Fractional delay filters can often be used as building blocks in various digital filters, especially adaptive or tunable filter structures.

Several FIR and allpass filter design techniques for approximating an arbitrary delay were discussed. The target response corresponds to the well-known sampled and shifted sinc function. The maximally-flat approximations are easy to use, since the coefficients can be expressed in closed form for both FIR and allpass filters. These techniques are called Lagrange interpolation and the Thiran allpass filter, respectively. Least squares design methods require solving a matrix equation but offer flexibility since the approximation bandwidth can be specified and a frequency-dependent error weighting function can be used. Equiripple approximation methods yield a filter with optimal properties in the minimax sense. In the case of allpass filters, the optimization criterion may be based on either the phase or the phase delay function.

In many cases, the fractional delay element must be time-variant in the sense that the delay parameter changes over time. Methods for implementing time-varying fractional delay filters were described. For FIR filters, the Farrow structure is a good choice. Since allpass filters are recursive, transients may cause problems in time-varying situations. Examples of transient elimination using a recently developed technique were given.

Research on the design of FD filters and their applications is currently in a very active phase, which is demonstrated by the fact that more than 80 contributions have been published after the earlier review article [71]. The research has been active, e.g., in the fields of one- and multidimensional digital waveguides, synchronization of digital receivers, and time-delay estimation.

The bandlimited interpolation theory of one-dimensional functions is well established and well known so that the research focus is shifting from re-inventing the Lagrange interpolation to ever-diversifying fields of applications and efficient implementation structures. Another trend seems to be the interpolation in more than one dimensions, such as modeling of multidimensional differential equations, where fractional delay filters can provide help to reduce the often extremely heavy computational requirements involved.

9. ACKNOWLEDGMENTS

The work of the first author has been financed by the Academy of Finland. The authors are grateful to Mr. Matti Airas, Dr. Gerry Cain, Dr. Paulo Diniz, Dr. Ewa Hermanowicz, Dr. Farokh Marvasti, Dr. Markku Renfors, Dr. Tapio Saramäki, Dr. Julius Smith, Dr. Andrzej Tarczynski, Mr. Tero Tolonen, and Dr. Jussi Vesma for helpful discussions and criticism.

10. MATLAB FILES

The MATLAB files listed below were used to produce the signal figures in this chapter. MATLAB version 5.2 for Windows was used. The MATLAB Signal Processing ToolBox (version 4.1 was used) is required to run these programs.

Table 3. MATLAB files used to produce the figures of this chapter.

Figure(s)	MATLAB M file	Uses functions
1	IDFDCHAR.M	-
2	SHFTSINC.M	-
4, 5	POLYFIGS.M	-
6, 7	LSFDFIGS.M	-
8(a), (b), (c), (d)	LAGRFIGS.M	HLAGR2.M
9(a), (b), (c), (d)	OETKENFIGS.M	INIHEQ2.M, HEQRIP2.M
11, 12	THIRFIGS.M *	APFLAT2.M
13	DISCONT.M	APFLAT2.M
18	APNOSTUP3.M	APFLAT2.M
19	APNSTUP3.M	APFLAT2.M
20	NUPOLFIG.M	EXPOLF1.M, POLF1.M, POLJIT.M
26	DOPPLERD.M	-
28	ARRCOMP1.M	BPIARR.M, FINTARR2.M
29	FDCOMB.M	-

* Note: You must run LAGRFIGS.M before THIRFIGS.M in order to be able to produce Figures 12(a) and 12(b).

11. REFERENCES

The following keywords have been found useful when searching for literature on this topic: fractional delay, fractional sample delay, interpolation filter, phase shifter, polynomial interpolation, and resampling.

[1] J. W. Adams, “Alternate approach to digital phase shift filters,” in *Proc. Int. Symp. Signal Process.: Theories, Implementations and Applications*, Brisbane, Australia, Aug. 1987, pp. 160–165.

[2] R. Adams and T. Kwan, “Theory and VLSI architectures for asynchronous sample-rate converters,” *J. Audio Eng. Soc*., vol. 41, no. 7/8, pp. 539–555, Jul/Aug. 1993.

[3] M. Akamine, K. Miseki, and M. Oshikiri, “Improvement of ADP-CELP speech coding at 4 kbit/s,” in *Proc. Global Telecommun. Conf.*, Phoenix, AZ, Dec. 1991, vol. 3, pp. 1869–1873.

[4] A. Aldroubi, M. Unser, and M. Eden, "Cardinal spline filters: stability and convergence to the ideal sinc interpolator," *Signal Process.*, vol. 28, no. 2, Aug 1992, pp. 127–138.

[5] J. Armstrong and D. Strickland, "Symbol synchronization using signal samples and interpolation," *IEEE Trans. Commun.*, vol. 41, no. 2, pp. 318–321, Feb. 1993.

[6] A. Barwicz, D. Bellemare, and R. Z. Morawski, "Digital correction of A/D conversion error due to multiplexing delay," *IEEE Trans. Instr. Meas.*, vol. 39, no. 1, pp. 76–79, Feb. 1990.

[7] M. G. Bellanger, G. Bonnerot, and M. Coudreuse, "Digital filtering by polyphase network: application to sample-rate alteration and filter banks," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 24, no. 2, pp. 109–114, Apr. 1976.

[8] C. Bertrand and P. Sehier, "A novel approach for full digital modems implementing asynchronous sampling techniques," in *Proc. IEEE Global Telecommun. Conf.*, London, UK, Nov. 1996, vol. 2, pp. 1320–1324.

[9] G. Braileanu, "Digital filters with implicit interpolated output," *IEEE Trans. Signal Process.*, vol. 45, no. 10, pp. 2551–2560, Oct. 1997.

[10] R. Bristow-Johnson, "Wavetable synthesis 101, a fundamental perspective," presented at the *Audio Eng. Soc. 101st Conv.*, Los Angeles, CA, Nov. 1996, preprint no. 4400.

[11] H. Brandenstein and R. Unbehauen, "Fractional delay FIR filter design with the Darlington transform," *IEEE Trans. Circ. Syst.—Part II*, vol. 46, no. 1, pp. 208–211, Jan. 1999.

[12] G. D. Cain, N. P. Murphy, and A. Tarczynski, "Evaluation of several FIR fractional-sample delay filters," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Adelaide, Australia, Apr. 1994, vol. 3, pp. 621–624.

[13] G. D. Cain, A. Yardim, and P. Henry, "Offset windowing for FIR fractional-sample delay," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Process.*, Detroit, MI, May 1995, vol. 2, pp. 1276–1279.

[14] J. M. de Carvalho and J. V. Hanson, "Efficient sampling rate conversion with cubic splines," in *Proc. SBT/IEEE Int. Telecommun. Symp.*, Sept. 1990, pp. 439–442.

[15] H. Chen, W. C. Wong, and C. C. Ko, "Comparison of pitch prediction and adaptation algorithms in forward and backward adaptive CELP systems," *IEE Proc.—I: Commun., Speech and Vision*, vol. 140, no. 4, pp. 240–245, Aug. 1993.

[16] A. J. R. M. Coenen, "Novel generalized optimal fractional delay filter design for navigational purposes," in *Proc. 9th Int. Symp. Personal Indoor and Mobile Radio Commun.*, Boston, MA, Sept. 1998, pp. 481–485.

[17] P. R. Cook, "Integration of physical modeling for synthesis and animation," in *Proc. Int. Computer Music Conf.*, Banff, Canada, Sept. 1995, pp. 525–528.

[18] R. E. Crochiere, L. R. Rabiner, and R. R. Shively, "A novel implementation of digital phase shifters," *Bell Syst. Tech. J.*, vol. 54, no. 8, pp. 1497–1502, Oct. 1975.

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

[20] S. Cucchi, F. Desinan, G. Parladori, and G. Sicuranza, "DSP implementation of arbitrary sampling frequency conversion for high quality sound application," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Toronto, ON, Canada, May 1991, vol. 5, pp. 3609–3612.

[21] R. B. Dannenberg, "Interpolation error in waveform table lookup," in *Proc. Int. Computer Music Conf.*, Ann Arbor, MI, Oct. 1998, pp. 240–243.

[22] J. Dattorro, "Effect design, part 2: delay-line modulation and chorus," *J. Audio Eng. Soc.*, vol. 45, no. 10, pp. 764–788, Oct. 1997.

[23] A. G. Dempster and N. P. Murphy, “Lagrange interpolator filters and binomial windows,” *Signal Process.*, vol. 76, pp. 81–91, Jul. 1999.
[24] A. G. Dempster and N. P. Murphy, “Efficient interpolators and filter banks using multiplier blocks,” accepted for publication in IEEE Trans. Signal Process., 1999.
[25] P. S. R. Diniz, R. Merched, and M. Petraglia, “A new delayless subband adaptive filter structure,” in *Proc. IEEE Int. Symp. Circ. Syst.*, Monterey, CA, May 1998, vol. 5, pp. 162–165.
[26] N. A. Dodgson, “Quadratic interpolation for image resampling,” *IEEE Trans. Image Process.*, vol. 6, no. 9, pp. 1322–1326, Sept. 1997.
[27] S. R. Dooley and A. K. Nandi, “Fast frequency estimation and tracking using Lagrange interpolation,” *Electron. Lett.*, vol. 34, no. 20, pp. 1908–1910, Oct. 1998.
[28] S. R. Dooley and A. K. Nandi, “On explicit time delay estimation using the Farrow structure,” *Signal Process.*, vol. 72, no. 1, pp. 53–57, Jan. 1999.
[29] S. R. Dooley and A. K. Nandi, “Adaptive subsample time delay estimation using Lagrange interpolators,” *IEEE Signal Process. Lett.*, vol. 6, no. 3, pp. 65–67, Mar. 1999.
[30] K. Egiazarian, T. Saramäki, H. Chugurian, and J. Astola, “Modified B-spline interpolators and filters: synthesis and efficient implementation,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Atlanta, GA, May 1996, vol. 3, pp. 1743–1746.
[31] L. Erup, F. M. Gardner, and R. A. Harris, “Interpolation in digital modems—part II: implementation and performance,” *IEEE Trans. Commun.*, vol. 41, no. 6, pp. 998–1008, June 1993.
[32] C. W. Farrow, “A continuously variable digital delay element,” in *Proc. IEEE Int. Symp. Circ. Syst.*, Espoo, Finland, June 1988, vol. 3, pp. 2641–2645.
[33] P. Fernández-Cid and F. J. Casajús-Quirós, “Enhanced quality and variety for chorus/flange units,” in *Proc. COST-G6 Digital Audio Effects Workshop*, Barcelona, Spain, Nov. 1998, pp. 35–39.
[34] A. Fettweis, “A simple design of maximally flat delay digital filters,” *IEEE Trans. Audio and Electroacoust.*, vol. 20, no. 2, pp. 112–114, June 1972.
[35] D. Fu and A. N. Willson, Jr., “Interpolation in timing recovery using a trigonometric polynomial and its implementation,” in *Proc. IEEE Global Telecommun. Conf.*, Sydney, Australia, Nov. 1998, pp. 173–178.
[36] F. M. Gardner, “Interpolation in digital modems—part I: fundamentals,” *IEEE Trans. Commun.*, vol. 41, no. 3, pp. 501–507, Mar. 1993.
[37] F. M. Gardner, “Difficulties with fractional-delay filters,” *IEEE Signal Process. Mag.*, vol. 13, no. 4, p. 16, July 1996.
[38] W. G. Gardner, *3-D Audio Using Loudspeakers*. Doctoral dissertation, Massachusetts Institute of Technology, Cambridge, MA, Sept. 1997.
[39] M. C. Gill and L. P. Sabel, “On the use of interpolation in digital demodulators,” *Australian Telecommun. Review*, vol. 27, no. 2, pp. 25–32, 1993.
[40] S. J. Godsill, “Recursive restoration of pitch variation defects in musical recordings,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Adelaide, Australia, Apr. 1994, vol. 2, pp. 233–236.
[41] S. J. Godsill and P. J. W. Rayner, *Digital Audio Restoration*. London, UK: Springer Verlag, 1998.
[42] M. M. Goodwin and G. W. Elko, “Constant beamwidth beamforming,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Minneapolis, MN, Apr. 1993, vol. 1, pp. 169–172.

[43] f. harris, “Performance and design of Farrow filter used for arbitrary resampling,” in *Proc. 13th Int. Conf. Digital Signal Process.*, July 1997, vol. 2, pp. 595–599.

[44] f. harris, “Performance and design considerations of the Farrow filter when used for arbitrary resampling of sampled time series,” in *Proc. 31st Asilomar Conf. Signals, Syst., Computers*, Pacific Grove, CA, Nov. 1997, vol. 2, pp. 1745–1749.

[45] f. harris, I. Gurantz, and S. Tzukerman, “Digital T/2 Nyquist filtering using recursive all-pass two-stage resampling filters for a wide range of selectable signalling rates,” in *Proc. 26th Asilomar Conf. Signals, Syst., Computers*, Pacific Grove, CA, Oct. 1992, vol. 2, pp. 676–680.

[46] W. M. Hartmann, “Digital waveform generation by fractional addressing,” *J. Acoust. Soc. Am.*, vol. 82, no. 6, pp. 1883–1891, Dec. 1987.

[47] E. Hermanowicz, “Explicit formulas for weighting coefficients of maximally flat tunable FIR delayers,” *Electron. Lett.*, vol. 28, no. 20, pp. 1936–1937, Sept. 1992.

[48] E. Hermanowicz, “Weighted Lagrangian interpolating FIR filter,” in *Proc. European Signal Process. Conf.*, Trieste, Italy, Sept. 1996, vol. 2, pp. 1203–1206.

[49] E. Hermanowicz and M. Rojewski, “Design of FIR first order digital differentiators of variable fractional sample delay using maximally flat error criterion,” *Electron. Lett.*, vol. 30, no. 1, pp. 17–18, Jan. 1994.

[50] E. Hermanowicz, M. Rojewski, G. D. Cain, and A. Tarczynski, “Special discrete-time filters having fractional delay,” *Signal Process.*, vol. 67, no. 3, pp. 279–289, June 1998.

[51] F. B. Hildebrand, *Introduction to Numerical Analysis. Second Edition.* New York: McGraw-Hill, 1974. Also: New York: Dover, 1989, 669 p.

[52] H. S. Hou and H. C. Andrews, “Cubic splines for image interpolation and digital filtering,” *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 26, no. 6, pp. 508–517, Dec. 1978.

[53] L. B. Jackson, *Digital Filters and Signal Processing. Third Edition*. Norwell, MA: Kluwer Academic Publishers, 1996, 502 p.

[54] D. Jaffe and J. O. Smith, “Extensions of the Karplus-Strong plucked string algorithm,” *Computer Music J.*, vol. 7, no. 2, pp. 56–69, 1983.

[55] A. J. Jerri, “The Shannon sampling theorem—its various extensions and applications: a tutorial review,” *Proc. IEEE*, vol. 65, no. 11, pp. 1565–1596, Nov. 1977.

[56] J.-M. Jot, V. Larcher, and O. Warusfel, “Digital signal processing issues in the context of binaural and transaural stereophony,” presented at *the Audio Eng. Soc. 98th Conv.*, Paris, France, Feb. 1995, preprint no. 3980.

[57] J.-M. Jot, S. Wardle, and V. Larcher, “Approaches to binaural synthesis,” presented at the *Audio Eng. Soc. 105th Conv.*, San Francisco, CA, Sept. 1998, preprint no. 4861.

[58] L. J. Karam and J. H. McClellan, “Complex Chebyshev approximation for FIR filter design,” *IEEE Trans. Circ. Syst.—Part II*, vol. 42, no. 3, pp. 207–216, Mar. 1995.

[59] M. Karjalainen and U. K. Laine, “A model for real-time sound synthesis of guitar on a floating-point signal processor,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Toronto, ON, Canada, May 1991, vol. 5, pp. 3653–3656.

[60] S. Kay, “Some results in linear interpolation theory,” *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 31, no. 3, pp. 746–749, June 1983.

[61] D. Kim, M. J. Narasimha, and D. C. Cox, “Design of optimal interpolation filter for symbol timing recovery,” *IEEE Trans. Commun.*, vol. 45, no. 7, pp. 877–884, July 1997.

[62] P. Knutson, D. McNeely, and K. Horlander, “An optimal approach to digital raster mapper design,” *IEEE Trans. Consumer Electron.*, vol. 37, no. 4, pp. 746–752, Sept.

1991.

[63] C. C. Ko and Y. C. Lim, "Approximation of a variable-length delay line by using tapped delay line processing," *Signal Process.*, vol. 14, no. 4, pp. 363–369, June 1988.

[64] P. J. Kootsookos and R. C. Williamson, "FIR approximation of fractional sample delay systems," *IEEE Trans. Circ. Syst.—Part II*, vol. 43, no. 3, pp. 269–271, Mar. 1996.

[65] P. Kroon and B. S. Atal, "On the use of pitch predictors with high temporal resolution," *IEEE Trans. Signal Process.*, vol. 39, no. 3, pp. 733–735, Mar. 1991.

[66] T. I. Laakso, L. P. Sabel, A. Yardim, and G. D. Cain, "Optimal receive filters for the suppression of ISI and adjacent channel interference," *Electron. Lett.*, vol. 32, no. 15, pp. 1346–1347, July 1996.

[67] T. I. Laakso, T. Saramäki, and G. D. Cain, "Asymmetric Dolph–Chebyshev, Saramäki, and transitional windows for fractional delay FIR filter design," in *Proc. 38th Midwest Symp. Circ. Syst.*, Rio de Janeiro, Brazil, Aug. 1995, vol. 1, pp. 580–583.

[68] T. I. Laakso, A. Tarczynski, N. P. Murphy, and V. Välimäki, "Polynomial filtering approach to reconstruction and noise reduction of nonuniformly sampled signals," *Signal Process.*, vol. 80, 2000.

[69] T. I. Laakso and V. Välimäki, "Energy-based effective length of the impulse response of a recursive filter," *IEEE Trans. Instr. Meas.*, vol. 47, no. 6, pp. 7–17, Feb. 1999.

[70] T. I. Laakso, V. Välimäki, and J. Henriksson, "Tunable downsampling using fractional delay filters with applications to digital TV transmission," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Detroit, MI, May 1995, vol. 2, pp. 1304–1307.

[71] 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, no. 1, pp. 30–60, Jan. 1996.

[72] U. K. Laine, "Digital modelling of a variable-length acoustic tube," in *Proc. Nordic Acoustical Meeting*, Tampere, Finland, June 1988, pp. 165–168.

[73] U. Lambrette, K. Langhammer, and H. Meyr, "Variable sample rate digital feedback NDA timing synchronization," in *Proc. IEEE Global Telecommun. Conf.*, London, UK, Nov. 1996, vol. 2, pp. 1348–1352.

[74] M. Lang, "Allpass filter design and applications," *IEEE Trans. Signal Process.*, vol. 46, no. 9, pp. 2505–2514, Sept. 1998.

[75] M. Lang and T. I. Laakso, "Simple and robust method for the design of allpass filters using least-squares phase error criterion," *IEEE Trans. Circ. Syst.—Part II*, vol. 41, no. 1, pp. 40–48, Jan. 1994.

[76] S.–H. Leung and C. W. Barnes, "State-space realization of fractional-step delay digital filters with applications to array beamforming," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 32, no. 2, pp. 317–380, Apr. 1984.

[77] Y. C. Lim, J.–H. Lee, C. K. Chen, and R.–H. Yang, "A weighted least squares algorithm for quasi-equiripple FIR and IIR digital filter design," *IEEE Trans. Signal Process.*, vol. 40, no. 3, pp. 551–558, Mar. 1992.

[78] T. J. Lim and M. D. Macleod, "On-line B-spline and Lagrange interpolation," Technical report CUED/F-INFENG/TR 202, Univ. of Cambridge, UK, Dec. 1994.

[79] G.–S. Liu and C.–H. Wei, "Programmable fractional sample delay filter with Lagrange interpolation," *Electron. Lett.*, vol. 26, no. 19, pp. 1608–1610, Sept. 1990.

[80] G.–S. Liu and C.–H. Wei, "A new variable fractional sample delay filter with nonlinear interpolation," *IEEE Trans. Circ. Syst.—Part II*, vol. 39, no. 2, pp. 123–126, Feb. 1992.

[81] R. J. Marks II, *Introduction to Shannon Sampling and Interpolation Theory*. New York, NY: Springer Verlag, 1991.

[82] J. S. Marques, J. M. Tribolet, I. M. Trancoso, and L. B. Almeida, “Pitch prediction with fractional delays in CELP coding,” in *Proc. European Conf. Speech Commun. and Tech.*, Paris, France, Sept. 1989, vol. 2, pp. 509–512.

[83] J. S. Marques, I. M. Trancoso, J. M. Tribolet, and L. B. Almeida, “Improved pitch prediction with fractional delays in CELP coding,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Albuquerque, NM, April 1990, vol. 2, pp. 665–668.

[84] F. Marvasti, *A Unified Approach to Zero-Crossings and Nonuniform Sampling*. Oak Park, IL: Nonuniform Publishing, 1987.

[85] F. Marvasti, M. Analoui, and M. Gamshadzahi, “Recovery of signals from nonuniform samples using iterative methods,” *IEEE Trans. Signal Process.*, vol. 39, no. 4, pp. 872–878, Apr. 1991.

[86] D. Massie, “Wavetable sampling synthesis,” in M. Kahrs and K. Brandenburg (eds.), *Applications of Digital Signal Processing to Audio and Acoustics*. Norwell, MA: Kluwer, 1998, pp. 311–341.

[87] N. E. Mastorakis, “A new approach for the fractional sample delay filters,” in *Proc. Third IEEE Int. Conf. Electron., Circ. Syst.*, vol. 1, pp. 502–505, Oct. 1996.

[88] S. McClellan, J. D. Gibson, and B. K. Rutherford, “Efficient pitch filter encoding for variable rate speech processing,” *IEEE Trans. Audio and Speech Process.*, vol. 7, no. 1, pp. 18–29, Jan. 1999.

[89] Y. Medan, “Using super resolution pitch in waveform speech coders,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Toronto, ON, Canada, May 1991, vol. 1, pp. 633–636.

[90] U. Mengali and A. N. D’Andrea, *Synchronization Techniques for Digital Receivers.* New York, NY: Plenum Press, 1997.

[91] R. Merched, P. S. R. Diniz, and M. Petraglia, “A delayless alias-free subband adaptive filter structure,” in *Proc. IEEE Int. Symp. Circ. Syst.*, Hong Kong, June 1997, vol. 4, pp. 2329–2332.

[92] R. Merched, P. S. R. Diniz, and M. Petraglia, “A delayless alias-free subband adaptive filter structure,” *IEEE Trans. Signal Process.*, vol. 47, no. 6, pp. 1580–1591, June 1999.

[93] H. Meyr and G. Ascheid, *Synchronization in Digital Communications, Vol. 1*. New York: Wiley, 1990.

[94] H. Meyr, M. Moeneclaey, and S. A. Fechtel, *Digital Communication Receivers—Synchronization, Channel Estimation, and Signal Processing*. New York: Wiley, 1998.

[95] S. Minocha, S. C. Dutta Roy, and B. Kumar, “A note on the FIR approximation of a fractional sample delay,” *Int. J. Circ. Theor. Appl.*, vol. 21, no. 3, pp. 265–274, May–June 1993.

[96] F. Mintzer, “On half-band, third-band, and *N*th-band FIR filters and their design,” *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 30, pp. 734–738, Oct. 1982.

[97] R. Msallam, S. Dequidt, S. Tassart, and R. Caussé, “Physical model of the trombone including nonlinear propagation effects,” in *Proc. Int. Symp. Musical Acoust.*, Edinburgh, Scotland, Aug. 1997, vol. 2, pp. 419–424.

[98] F. R. Moore, “Table lookup noise for sinusoidal digital oscillators,” *Computer Music J.*, vol. 1, no. 2, pp. 26–29, 1977. Reprinted in C. Roads and J. Strawn, *Foundations of Computer Music*. Cambridge, MA: MIT Press, 1985, pp. 326–334.

[99] D. H. Mugler, “Linear prediction of a bandlimited signal from past samples at arbitrary points: an SVD-based approach,” in *Proc. Workshop Sampling Theory and Appl.*, Riga, Latvia, Sept. 1995, pp. 113–118.

[100] N. P. Murphy, A. Krukowski, and I. Kale, “Implementation of wideband integer and fractional delay element,” *Electron. Lett.*, vol. 30, no. 20, pp. 1658–1659, Sept. 1994.
[101] N. P. Murphy, A. Krukowski, and A. Tarczynski, “An efficient fractional sample delayer for digital beam steering,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Munich, Germany, Apr. 1997, vol. 3, pp. 2245–2248.
[102] N. P. Murphy, T. I. Laakso, and G. Allen, “Fast resampling of nonuniformly sampled signals using interpolative FIR filters,” in *Proc. Workshop Sampling Theory and Appl.*, Riga, Latvia, Sept. 1995, pp. 241–246.
[103] N. P. Murphy, A. Tarczynski, and T. I. Laakso, “Sampling-rate conversion using a wideband tuneable fractional delay element,” in *Proc. Nordic Signal Process. Symp.*, Espoo, Finland, Sept. 1996, pp. 423–426.
[104] S. N. Nazra, “Linear phase IIR filter with equiripple stopband,” *IEEE Trans. Acoust. Speech, Signal Process.*, vol. 31, no. 3, pp. 744–746, June 1983.
[105] T. Q. Nguyen, T. I. Laakso, and R. D. Koilpillai, “Eigenfilter approach for the design of allpass filter approximating a given phase response,” *IEEE Trans. Signal Process.*, vol. 42, no. 9, pp. 2257–2263, Sept. 1994.
[106] G. Oetken, T. W. Parks, and H. W. Schüßler, “New results in the design of digital interpolators,” *IEEE Trans. Acoust. Speech, Signal Process.*, vol. 23, no. 3, pp. 301–309, June 1975.
[107] G. Oetken, “A new approach for the design of digital interpolating filters,” *IEEE Trans. Acoust. Speech, Signal Process.*, vol. 27, no. 6, pp. 637–643, Dec. 1979.
[108] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*. Englewood Cliffs, NJ: Prentice–Hall, 1989.
[109] D. O’Shea and S. McGrath, “A novel interpolation method for maximum likelihood timing recovery,” *Wireless Personal Commun.*, vol. 4, pp. 315–324, 1997.
[110] A. Papoulis, *Signal Analysis*. New York: McGraw-Hill, 1977.
[111] S. Park, G. Hillman, and R. Robles, “A novel structure for real-time digital sample-rate converters with finite precision error analysis,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Toronto, ON, Canada, May 1991, vol. 5, pp. 3613–3616.
[112] T. W. Parks and C. S. Burrus, *Digital Filter Design*. New York: Wiley, 1987.
[113] S.–C. Pei and C.–C. Tseng, “A comb filter design using fractional-sample delay,” *IEEE Trans. Circ. Syst.—Part II*, vol. 45, no. 6, pp. 649–653, June 1998.
[114] L. D. Philipp, A. Mahmood, and B. L. Philipp, “An improved refinable rational approximation to the ideal time delay,” *IEEE Trans. Circ. Syst.—Part II*, vol. 46, no. 5, pp. 637–640, May 1999.
[115] J. R. Pierce and S. A. Van Duyne, “A passive nonlinear digital filter design which facilitates physics-based sound synthesis of highly nonlinear musical instruments,” *J. Acoust. Soc. Am.*, vol. 101, no. 2, pp. 1120–1126, Feb. 1997.
[116] K. C. Pohlman, *Principles of Digital Audio. Third Edition*. New York: McGraw-Hill, 1995.
[117] K. Preuss, “On the design of FIR filters by complex approximation,” *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 37, no. 5, pp. 702–712, May 1989.
[118] R. G. Pridham and R. A. Mucci, “A novel approach to digital beamforming,” *J. Acoust. Soc. Am.*, vol. 63, no. 2, pp. 425–434, Feb. 1978.
[119] R. G. Pridham and R. A. Mucci, “Digital interpolation beamforming for lowpass and bandpass signals,” *Proc. IEEE*, vol. 67, pp. 904–919, June 1979.
[120] W. Putnam and J. O. Smith, “Design of fractional delay filters using convex optimization,” in *Proc. IEEE Workshop Appl. Signal Process. Audio and Acoustics*, New Paltz,

NY, Oct. 1997.
[121] M. F. Pyfer and R. Ansari, "The design and application of optimal FIR fractional-slope phase filters," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Dallas, TX, Apr. 1987, vol. 2, pp. 896–899.
[122] T. Ramstad, "Digital methods for conversion between arbitrary sampling frequencies," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 32, no. 3, pp. 577–591, June 1984.
[123] T. Ramstad, "Fractional rate decimator and interpolator design," in *Proc. European Signal Process. Conf.*, Rhodes, Greece, Sept. 1998, vol. 4, pp. 1948–1952.
[124] P. A. Regalia, "Special filter designs," in S. K. Mitra and J. F. Kaiser (eds.), *Handbook of Digital Signal Processing*. New York: Wiley, 1993, pp. 907–980.
[125] S. Ries and M. Roeckerath–Ries, "Combined matched filter/interpolator for digital receivers," in *Proc. European Signal Process. Conf.*, Trieste, Italy, Sept. 1996, vol. 1, pp. 627–630.
[126] D. Rocchesso, "Fractionally-addressed delay lines," in *Proc. COST-G6 Digital Audio Effects Workshop*, Barcelona, Spain, Nov. 1998, pp. 40–43.
[127] L. P. Sabel, "On the BER performance of digital demodulators using fixed point FIR interpolators," in *Proc. 3rd Annual Int. Conf. Universal Personal Commun.*, Sept./Oct. 1994, pp. 215–219.
[128] L. P. Sabel, T. I. Laakso, A. Yardim, and G. D. Cain, "Improved delay root-Nyquist filters for symbol synchronisation in PCS receivers," in *Proc. IEEE Global Telecommun. Conf.*, Singapore, Nov. 1995, vol. 2, pp. 1302–1306.
[129] L. P. Sabel, A. Yardim, G. D. Cain, and T. I. Laakso, "Effects of delay-root-Nyquist filters on the performance of digital demodulators," in *Proc. Int. Conf. Telecommun.*, Bali, Indonesia, Apr. 1995, pp.75–79.
[130] S. Samadi and H. Iwakura, "Variable Taylor realization and improved approximation of FIR fractional delay systems," *Int. J. Circ. Theor. Appl.*, vol. 26, no. 5, pp. 513–522, Sept–Oct. 1998.
[131] T. Saramäki, "Finite impulse response filter design," in S. K. Mitra and J. F. Kaiser (eds.), *Handbook of Digital Signal Processing*. New York: Wiley, 1993, pp. 155–277.
[132] T. Saramäki and M. Renfors, "A Remez-type algorithm for designing digital filters composed of all-pass sections based on phase approximations," in *Proc. 38th Midwest Symp. Circ. Syst.*, Rio de Janeiro, Brazil, Aug. 1995, vol. 1, pp. 571–575.
[133] T. Saramäki and M. Renfors, "*N*th-band filter design," in *Proc. European Signal Process. Conf.*, Rhodes, Greece, Sept. 1998, vol. 4, pp. 1943–1947 .
[134] T. Saramäki and T. Ritoniemi, "An efficient approach for conversion between arbitrary sampling frequencies," in *Proc. Int. Symp. Circ. Syst.*, Atlanta, GA, May 1996, vol. 2, pp. 285–288.
[135] L. Savioja, "Improving the three-dimensional digital waveguide mesh by interpolation," in *Proc. Nordic Acoust. Meeting*, Stockholm, Sweden, Sept. 1998, pp. 265–268.
[136] L. Savioja, J. Huopaniemi, T. Lokki, and R. Väänänen, "Creating interactive virtual acoustic environments," *J. Audio Eng. Soc.*, vol. 47, no. 9, pp. 675–705, Sept. 1999.
[137] L. Savioja and V. Välimäki, "The bilinearly deinterpolated waveguide mesh," in *Proc. Nordic Signal Process. Symp.*, Espoo, Finland, Sept. 1996, pp. 443–446.
[138] L. Savioja and V. Välimäki, "Improved discrete-time modeling of multi-dimensional wave propagation using the interpolated digital waveguide mesh," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Munich, Germany, Apr. 1997, vol. 1, pp. 459–462.

[139] L. Savioja and V. Välimäki, "Reduction of the dispersion error in the interpolated digital waveguide mesh using frequency warping," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Phoenix, AZ, Mar. 1999, vol. 2, pp. 973–976.

[140] L. Savioja and V. Välimäki, "Reducing the dispersion error in the digital waveguide mesh using interpolation and frequency-warping techniques," *IEEE Trans. Speech and Audio Process.*, vol. 8, no. 2, Mar. 1999.

[141] G. Scavone and P. R. Cook, "Real-time computer modeling of woodwind instruments," in *Proc. Int. Symp. Musical Acoust.*, Leavenworth, WA, June/July 1998.

[142] R. W. Schafer and L. R. Rabiner, "A digital signal processing approach to interpolation," *Proc. IEEE*, vol. 61, no. 6, pp. 692–702, June 1973.

[143] D. W. E. Schobben and P. C. W. Sommen, "Increasing beamsteering directions using polyphase decomposition," in *Proc. 8th IEEE Signal Process. Workshop on Statistical Signal and Array Process.*, June 1996, pp. 117–120.

[144] M. Schulist, "Improvements of a complex FIR filter design algorithm," *Signal Process.*, vol. 20, no. 1, pp. 81–90, May 1990.

[145] I. W. Selesnick, "Lowpass filters realizable as all-pass sums: design via a new flat delay filter," *IEEE Trans. Circ. Syst.—Part II*, vol. 46, no. 1, pp. 40–50, Jan. 1999.

[146] S. Signell, "Design of maximally flat group delay discrete-time recursive filters," in *Proc. IEEE Int. Symp. Circ. Syst.*, Montreal, QC, Canada, May 1984, vol. 1, pp. 192–196.

[147] S. Sivanand, J.–F. Yang, and M. Kaveh, "Focusing filters for wide-band direction finding," *IEEE Trans. Signal Process.*, vol. 39, no. 2, pp. 437–445, Feb. 1991.

[148] S. Sivanand, "Variable-delay implementation using digital FIR filters," in *Proc. Int. Conf. Signal Process. Appl. Theor.*, Boston, MA, Nov. 1992, pp. 1238–1243.

[149] J. O. Smith, *Techniques for Digital Filter Design and System Identification with Application to the Violin*. Doctoral dissertation, Report STAN-M-14, CCRMA, Dept. of Music, Stanford University, Stanford, CA, June 1983.

[150] J. O. Smith, "Physical modeling using digital waveguides," *Computer Music J.*, vol. 16, no. 4, pp. 74–91, 1992. Available online at http://www-ccrma.stanford.edu/~jos/.

[151] J. O. Smith, "Physical modeling synthesis update," *Computer Music J.*, vol. 20, no. 2, pp. 44–56, 1996. Available online at http://www-ccrma.stanford.edu/~jos/.

[152] J. O. Smith, "Acoustic modeling using digital waveguides," in C. Roads, S. T. Pope, A. Piccialli, and G. De Poli (eds.), *Musical Signal Processing*. Lisse, the Netherlands: Swets & Zeitlinger, 1997, pp. 221–263. Available online at http://www-ccrma.stanford.edu/~jos/.

[153] J. O. Smith and B. Friedlander, "Adaptive interpolated time-delay estimation," *IEEE Trans. Aerospace and Electronic Syst.*, vol. 21, no. 3, pp. 180–199, Mar. 1985.

[154] J. O. Smith and P. Gossett, "A flexible sampling-rate conversion method," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, San Diego, CA, Mar. 1984, vol. 2, pp. 19.4.1–4.

[155] H. Strauss, "Implementing Doppler shifts for virtual auditory environments," presented at the *Audio Eng. Soc. 104th Conv.*, Amsterdam, the Netherlands, May 1998, preprint no. 4687.

[156] H. W. Strube, "Sampled-data representation of a nonuniform lossless tube of continuously variable length," *J. Acoust. Soc. Am.*, vol. 57, no. 1, pp. 256–257, Jan. 1975.

[157] H. W. Strube, "How to make an all-pass filter with a desired impulse response," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 30, no. 2, pp. 336–337, Apr. 1982.

[158] R. Sudhakar, R. C. Agarwal, and S. C. Dutta Roy, "Time domain interpolation using

differentiators," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 30, no. 6, pp. 992–997, Dec. 1982.

[159] C. S. Sullivan, "Extending the Karplus–Strong algorithm to synthesize electric guitar timbres with distortion and feedback," *Computer Music J.*, vol. 14, no. 3, pp. 26–37, 1990.

[160] A. Tarczynski and G. D. Cain, "Stabilisation of a class of feedback systems using fractional-sample delayors," *Applied Signal Processing*, vol. 1, no. 1, pp. 55–58, 1994.

[161] A. Tarczynski and G. D. Cain, "Design of IIR fractional-sample delay filters," in *Proc. 2nd Int. Symp. DSP for Commun. Syst.*, Adelaide, Australia, Apr. 1994.

[162] A. Tarczynski, G. D. Cain, E. Hermanowicz, and M. Rojewski, "WLS design of variable frequency response FIR filters," in *Proc. Int. Symp. Circ. Syst.*, Hong Kong, June 1997, vol. 4, pp. 2244–2247.

[163] A. Tarczynski, W. Kozinski, and G. D. Cain, "Sampling rate conversion using fractional-sample delay," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Adelaide, Australia, Apr. 1994, vol. 3, pp. 285–288.

[164] S. Tassart and Ph. Depalle, "Analytical approximation of fractional delays: Lagrange interpolators and allpass filters," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Munich, Germany, Apr. 1997, vol. 1, pp. 455–458.

[165] S. Tassart, R. Msallam, Ph. Depalle, and S. Dequidt, "A fractional delay application: time-varying propagation speed in waveguides," in *Proc. Int. Computer Music Conf.*, Thessaloniki, Greece, Sept. 1997, pp. 256–259.

[166] J.–P. Thiran, "Recursive digital filters with maximally flat group delay," *IEEE Trans. Circ. Theory*, vol. 18, no. 6, pp. 659–664, Nov. 1971.

[167] T. Tolonen, V. Välimäki, and M. Karjalainen, "Modeling of tension modulation nonlinearity in plucked strings," accepted for publication in IEEE Trans. Speech and Audio Process., 1999.

[168] V. Tuukkanen, J. Vesma, and M. Renfors, "Efficient near optimal maximum likelihood symbol timing recovery in digital modems," in *Proc. Int. Symp. on Personal, Indoor and Mobile Radio Commun.*, Helsinki, Finland, Sept. 1997, pp. 825–829.

[169] M. Unser, A. Aldroubi, and M. Eden, "Polynomial spline signal approximations: filter design and asymptotic equivalence with Shannon's sampling theorem," *IEEE Trans. Information Theory*, vol. 38, no. 1, pp. 95–103, Jan. 1992.

[170] M. Unser, A. Aldroubi, and M. Eden, "B-spline signal processing: Part I—theory," *IEEE Trans. Signal Process.*, vol. 41, no. 2, pp. 821–833, Feb. 1993.

[171] M. Unser, A. Aldroubi, and M. Eden, "B-spline signal processing: Part II—efficient design and applications," *IEEE Trans. Signal Process.*, vol. 41, no. 2, pp. 834–848, Feb. 1993.

[172] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[173] V. Välimäki, "A new filter implementation strategy for Lagrange interpolation," in *Proc. IEEE Int. Symp. Circ. Syst.*, Seattle, WA, Apr.–May 1995, vol. 1, pp. 361–364.

[174] V. Välimäki, *Discrete-Time Modeling of Acoustic Tubes Using Fractional Delay Filters*. Doctoral dissertation, Report no. 37, Helsinki University of Technology, Laboratory of Acoustics and Audio Signal Processing, Espoo, Finland, Dec. 1995. Available at http://www.acoustics.hut.fi/~vpv/publications/vesa_phd.html.

[175] V. Välimäki, J. Huopaniemi, M. Karjalainen, and Z. Jánosy, "Physical modeling of plucked string instruments with application to real-time sound synthesis," *J. Audio Eng. Soc.*, vol. 44, no. 5, pp. 331–353, May 1996.

[176] V. Välimäki and M. Karjalainen, "Implementation of fractional delay waveguide models using allpass filters," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Detroit, MI, May 1995, vol. 2, pp. 1524–1527.

[177] V. Välimäki, M. Karjalainen, and T. Kuisma, "Articulatory speech synthesis based on fractional delay waveguide filters," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Adelaide, Australia, Apr. 1994, vol. 1, pp. 585–588.

[178] V. Välimäki, M. Karjalainen, and T. I. Laakso, "Fractional delay digital filters," in *Proc. IEEE Int. Symp. Circ. Syst.*, Chicago, IL, May 1993, vol. 1, pp. 355–358.

[179] V. Välimäki, M. Karjalainen, and T. I. Laakso, "Modeling of woodwind bores with finger holes," in *Proc. Int. Computer Music Conf.*, Tokyo, Japan, Sept. 1993, pp. 32–39.

[180] V. Välimäki and T. I. Laakso, "Suppression of transients in variable recursive digital filters with a novel and efficient cancellation method," *IEEE Trans. Signal Process.*, vol. 46, no. 12, pp. 3408–3414, Dec. 1998.

[181] 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, Canada, Sept. 1995, pp. 327–334.

[182] V. Välimäki, T. Tolonen, and M. Karjalainen, "Signal-dependent nonlinearities for physical models using time-varying fractional delay filters," in *Proc. Int. Computer Music Conf.*, Ann Arbor, MI, Oct. 1998, pp. 264–267.

[183] V. Välimäki, T. Tolonen, and M. Karjalainen, "Plucked-string synthesis algorithms with tension modulation nonlinearity," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Phoenix, AZ, Mar. 1999, vol. 2, pp. 977–980.

[184] S. A. Van Duyne, D. A. Jaffe, G. P. Scandalis, and T. S. Stilson, "A lossless, click-free, pitchbend-able delay line loop interpolation scheme," in *Proc. Int. Computer Music Conf.*, Thessaloniki, Greece, Sept. 1997, pp. 252–255.

[185] D. Verdin and T. C. Tozer, "Interpolator filter structure for asynchronous timing recovery loops," *Electron. Lett.*, vol. 29, no. 5, pp. 490–492, Mar. 1993.

[186] J. Vesma, *Optimization and Applications of Polynomial-Based Interpolation Filters.* Doctoral dissertation, Tampere University of Technology, Tampere, Finland, May 1999.

[187] J. Vesma, M. Renfors, and J. Rinne, "Comparison of efficient interpolation techniques for symbol timing recovery," in *Proc. IEEE Global Telecommun. Conf.*, London, UK, Nov. 1996, vol. 2, pp. 953–957.

[188] J. Vesma and T. Saramäki, "Interpolation filters with arbitrary frequency response for all-digital receivers," in *Proc. Int. Symp. Circ. Syst.*, Atlanta, GA, May 1996, vol. 2, pp. 568–571.

[189] J. Vesma and T. Saramäki, "Optimization and efficient implementation of fractional delay FIR filters," in *Proc. Int. Conf. Electron., Circ. Syst.*, Rhodes, Greece, Oct. 1996, pp. 546–549.

[190] J. Vesma and T. Saramäki, "Optimization and efficient implementation of FIR filters with adjustable fractional delay," in *Proc. Int. Symp. Circ. Syst.*, Hong Kong, June 1997, vol. 4, pp. 2256–2259.

[191] J. Vesma, T. Saramäki, and M. Renfors, "Combined matched filter and polynomial-based interpolator for symbol synchronization in digital receivers," in *Proc. IEEE Int. Symp. Circ. Syst.*, Orlando, FL, May 1999, pp. 94–97.

[192] J. Vesma, V. Tuukkanen, and M. Renfors, "Maximum likelihood feedforward symbol timing recovery based on efficient interpolation techniques," in *Proc. Nordic Signal

Process. Symp., Espoo, Finland, Sept. 1996, pp. 183–186.

[193] D. Wise, J. Barish, and E. Lindemann, “DSP strategies for faster polynomial interpolation,” presented at the *Audio Eng. Soc. 101st Conv.*, Los Angeles, CA, Nov. 1996, preprint no. 4370.

[194] H.–K. Yang and M. Snelgrove, “Symbol timing recovery using oversampling techniques,” in *Proc. Int. Conf. Commun.*, Dallas, TX, June 1996, vol. 3, pp. 1296–1300.

[195] A. Yardim, G. D. Cain, and P. Henry, “Optimal two-term offset windowing for fractional delay,” *Electron. Lett.*, vol. 32, no. 6, pp. 526–527, Mar. 1996.

[196] A. Yardim, G. D. Cain, and A. Lavergne, “Performance of fractional-delay filters using optimal offset windows,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Munich, Germany, Apr. 1997, vol. 3, pp. 2233–2236.

[197] A. Yardim, L. J. Karam, J. H. McClellan, and G. D. Cain, “Performance of complex Chebyshev approximation in delay-root-Nyquist filter design,” in *Proc. IEEE Int. Symp. Circ. Syst*, Atlanta, GA, May 1996, vol. 2, pp. 169–172.

[198] A. Yardim, T. I. Laakso, L. P. Sabel, and G. D. Cain, “Design of efficient receive FIR filters for joint minimization of channel noise, ISI and adjacent channel interference,” in *Proc. IEEE Global Telecommun. Conf.*, London, UK, Nov. 1996, vol. 2, pp. 948–952.

[199] A. Yardim, T. I. Laakso, L. P. Sabel, and G. D. Cain, “Frequency-sampling design of delay-root-Nyquist filters,” *Electron. Lett.*, vol. 33, no. 1, pp. 35–37, Jan. 1997.

[200] L. H. Zetterberg and Q. Zhang, “Elimination of transients in adaptive filters with application to speech coding,” *Signal Process.*, vol. 15, no. 4, pp. 419–428, 1988.

[201] H. Zhang, “Interpolator for all-digital receivers,” *Electron. Lett.*, vol. 33, no. 4, pp. 261–262, Feb. 1996.

[202] H. Zhang and M. Renfors, “A study of performance of all-digital symbol synchronizers with polynomial interpolators,” in *Proc. Int. Conf. Commun. Tech.*, Beijing, China, May 1996, vol. 2, pp. 1013–1016.

[203] V. •ivojnovi• and H. Meyr, “Design of optimum interpolation filters for digital demodulators,” in *Proc. IEEE Int. Symp. Circ. Syst.*, Chicago, IL, May 1993, vol. 1, pp. 140–143.

[204] U. Zölzer and T. Boltze, “Interpolation algorithms: theory and application,” presented at the *Audio Eng. Soc. 97th Conv.*, San Francisco, CA, Nov. 1994, preprint no. 3898.

[205] U. Zölzer, *Digital Audio Signal Processing.* Chichester, UK: Wiley, 1997.