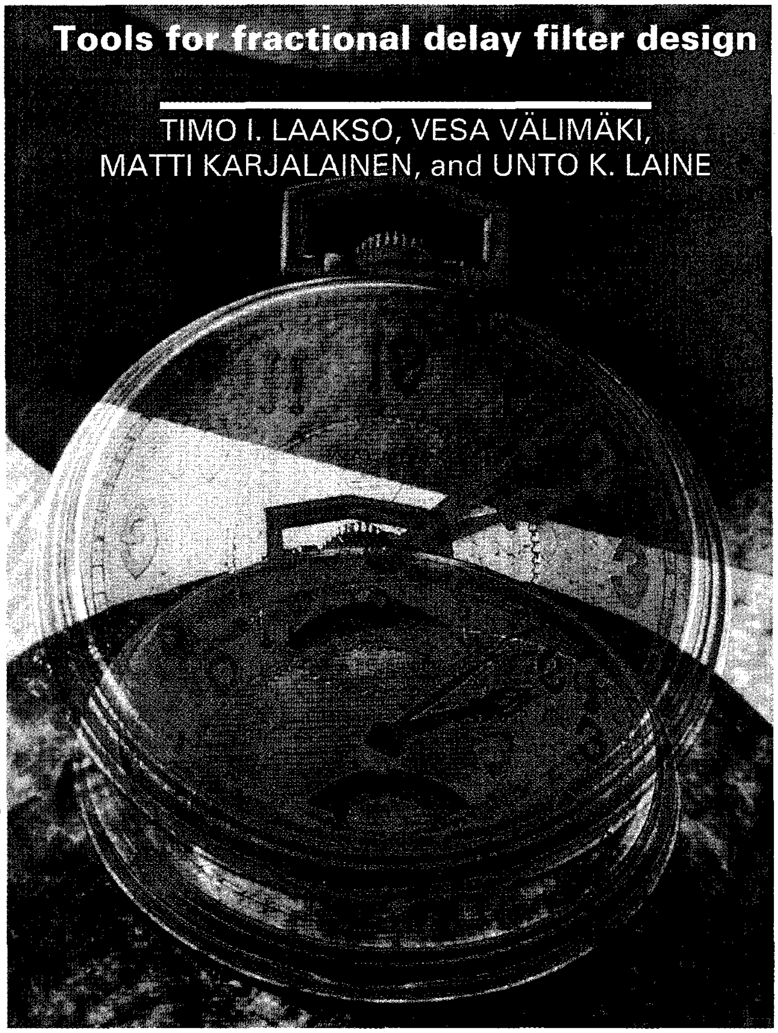
Splitting

Tools for fractional delay filter design

TIMO I. LAAKSO, VESA VÄLIMÄKI,
MATTI KARJALAINEN, and UNTO K. LAINE



the Unit Delay

A fractional delay filter is a device for bandlimited interpolation between samples. It finds applications in numerous fields of signal processing, including communications, array processing, speech processing, and music technology. In this article, we present a comprehensive review of FIR and allpass filter design techniques for bandlimited approximation of a fractional digital delay. Emphasis is on simple and efficient methods that are well suited for fast coefficient update or continuous control of the delay value. Various new approaches are proposed and several examples are provided to illustrate the performance of the methods. We also discuss the implementation complexity of the algorithms. We focus on four applications where fractional delay filters are needed: synchronization of digital modems, incommensurate sampling rate conversion, high-resolution pitch prediction, and sound synthesis of musical instruments.

1053-5888/96/$4.00©1996IEEE

Overview

One fundamental advantage of digital signal processing techniques over traditional analog methods is the easy implementation of a constant delay: the signal samples are simply stored in a buffer memory for the given time. This technique works perfectly as long as the desired delay is a multiple of the used sample interval. However, when a delay of a fraction of the sample interval is needed or, particularly, if it is desired to control the delay value continuously, more sophisticated methods must be used.

The problem of implementing a fractional delay (FD) by digital means occurs in several applications. In one of the first treatments on the subject [23], a digital phase shifter was proposed for three problems: echo cancellation, phased-array antenna processing, and pitch-synchronous synthesis of speech. Later papers have included applications such as time delay estimation [88, 104], null steering in the direction pattern of antenna arrays [48, 101, 102], timing adjustment and interpolation in digital modems [29, 66, 67, 30, 31, 33, 32, 7, 27, 136, 10, 95, 96, 72], sampling rate conversion systems [3, 112, 55], stabilization of feedback systems [111], speech coding [50, 51, 68, 70, 78], speech-assisted video processing [18, 19], sub-pixel interpolation [20, 47], modeling of the human vocal tract [108, 58, 122, 125, 128, 131], and modeling of musical instruments [39, 42, 110, 120, 121, 123, 127]. A comprehensive study of modeling of acoustic tubes using fractional delay filters has been presented in [126] and [135]. There are many potential applications in video processing, such as frame interpolation and sub-pixel interpolation [20].

As a consequence of the wide range of application areas, the results on approximation of a fractional delay are scattered in the literature and difficult to find. The proposed design techniques concentrate almost exclusively on finite-impulse-response (FIR) filters. Moreover, the approach in the majority of papers known to us is limited. The problem is viewed mainly as a time-domain interpolation problem, often leading to a modification of standard sampling rate conversion methods [23, 24, 2], or to the use of the traditional Lagrange interpolation technique [58, 66, 67, 37]. One of the few textbook treatments on FD filter design can be found in [93].

Here, we take a filter designer's point of view on the fractional delay problem. After formulating the (generally complex-valued) approximation problem in the frequency domain, we systematically review the well-established filter design theory and search for efficient solutions to this particular problem. Our article thus has a review nature and a serious effort has been made to find all the relevant literature addressing the problem. Besides giving a systematic presentation of previous results, we use the employed frequency-domain approach to give deeper insight to many methods and also to reveal useful design techniques that have previously not been considered for this problem.

Preliminaries

Notation and Concepts

Delaying a continuous-time signal $x_c(t)$ by an amount t_d is conceptually simple. A continuous-time ideal delay can be defined as a linear operator, L_c, which yields its output, $y_c(t)$ as

$$y_c(t) = L_c\{x_c(t)\} = x_c(t - t_D) \tag{1}$$

(Fig. 1a). Delaying of a uniformly sampled bandlimited (baseband) digital signal must be treated with care. When we simply convert Eq. 1 into discrete time by sampling at time instants $t = nT$, where n is an integer and T is the sampling interval, we obtain

$$y(n) = L\{x(n)\} = x(n - D) \tag{2}$$

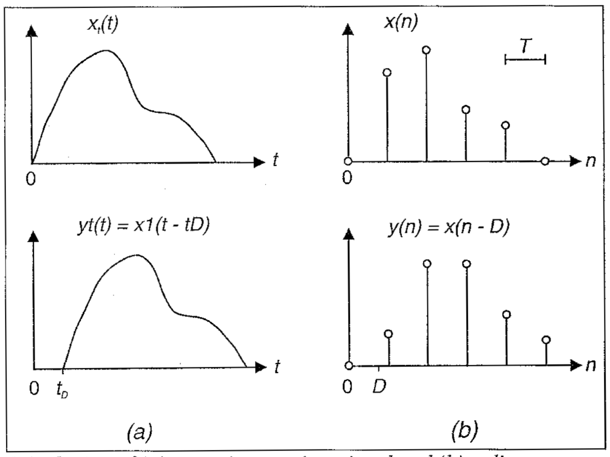
where D is a *positive real number* that can be split into the integer and fractional part as

$$D = \mathrm{Int}(D) + d \tag{3}$$

However, Eq. 2 is meaningful only for integer values of

Glossary of Notation

a_k	allpass filter coefficients	$h_{id}(n)$	impulse response of the ideal interpolator	$w(n)$	*time-domain window function*
$\mathbf{a}$	vector of allpass filter coefficients	$\mathbf{h}$	vector of FIR filter coefficients	$\mathbf{W}$	diagonal matrix with the elements of the window function $w(n)$
$A(z)$	allpass transfer function	$H(z)$	FIR transfer function	$W(\omega)$	frequency-domain weighting function
$c(n)$	coefficients of polynomial approximation	$H_{id}(z)$	the ideal transfer function	$x(n)$	filter input
$\mathbf{c}$	vector of cosine functions	L	FIR filter length ($L = N + 1$)	$y(n)$	filter output
$C(z)$	transfer function of polynomial approximation	n	discrete time index	α	passband width parameter ($0 < \alpha \leq 1$)
d	*fractional part of the total delay D*	N	filter order	$\beta(\omega)$	*phase error function in allpass filter design*
D	total noninteger delay to be approximated	M	net delay of the delay filter	δ	ripple parameter
$D(z)$	denominator of A(z)	P	order of spline or polynomial	$\Theta(\omega)$	phase response
$e(n)$	time-varying error sequence	$\mathbf{p}, \mathbf{P}$	vector and matrix in normal equations	Λ	diagonal matrix in group delay expression
$\mathbf{e}$	vector of complex exponentials	q	iteration index	$\tau_g(\omega)$	group delay response
E	general error measure	Q	number of polyphase branches	$\tau_p(\omega)$	phase delay response
$E(e^{j\omega})$	frequency-domain error function (complex-valued)	$r_{xx}(k)$	autocorrelation sequence of signal $x(n)$	ω	normalized angular frequency _
f	frequency variable	$\mathbf{s}$	vector of sine functions	ω_p	passband cutoff angular frequency
$\mathbf{G}$	matrix in group delay expression	$\mathbf{S}$	matrix of sine or sinc functions	ω_s	stopband cutoff angular frequency
$h(n)$	impulse response of an FIR filter	$S_{xx}(\omega)$	power spectrum of signal x(n)	Ω_k	set of values of ω
		T	sampling interval		
		$\mathbf{V}$	Vandermonde matrix		



1. Delaying of (a) a continuous-time signal and (b) a discrete-time signal.

D. In that case, the output sample is one of the previous signal samples, but for noninteger values of D, the output value would lie somewhere between two samples, which is impossible (Fig. 1b). Instead, the appropriate values on the sampling grid must be found via *bandlimited interpolation* (the problem of approximating negative values of D calls for *extrapolation* or *prediction* algorithms, which are beyond the scope of this article).

The problem can be solved by viewing a delay as a *resampling process*. The desired solution can be obtained by first reconstructing the continuous bandlimited signal and then resampling it after shifting [30, 31]. The task is thus related to interpolation in multirate filter design techniques [23, 24, 118, 119] or sampling rate conversion in general [5, 6, 9, 91, 92, 112, 55]. Note, however that our basic constraint is to keep the sampling rate unchanged.

As in multirate applications, we need not perform reconstruction and resampling explicitly, but they can be reduced to appropriate linear filtering operating at the chosen sampling rate. A key issue is how to formulate the problem such that well-advanced filter design theory can be utilized in an efficient manner.

Like any linear time-invariant operation, delaying can equivalently be considered in a suitable transform domain. The z-domain transfer function of the system of Fig. 2 is obtained formally as

$$H_{id}(z) = \frac{Y(z)}{X(z)} = \frac{z^{-D}X(z)}{X(z)} = z^{-D} \tag{4}$$

where $X(z) = Z\{s(n)\}$ and $Y(z)$ are the z-transforms of $x(n)$ and $y(n)$, respectively, and the subscript 'id' will stand for the desired (ideal) response. In Eq. 4, we have employed the following property of the z-transform:

$$Z\{x(n-D)\} = z^{-D}X(z) \tag{5}$$

which, strictly speaking, holds for integer values of D only. The term z^{-D} represents precisely the ideal filter of Fig. 2 in the z-domain, which performs the desired bandlimited delay operation at the used sampling rate. Clearly, z^{-D} cannot be realized exactly for noninteger D, but it must be approximated in some way.

One approach for approximation is to construct a series expansion for z^{-D} as proposed in [90]. However, a more general and fruitful approach is to formulate the design objective in the frequency domain. In many applications the specifications are easier to give in the frequency domain, and numerous design techniques are available. The frequency response (Fourier transform) of the delaying system of Fig. 2 is obtained from Eq. 4 by setting $z = e^{j\omega}$

$$H_{id}(e^{j\omega}) = e^{-j\omega D} \tag{6}$$

where $\omega=2\pi fT$ is the normalized angular frequency, and T is the sample interval. The desired frequency response is thus a complex-valued function that specifies both the magnitude and the phase response as

$$|H_{id}(e^{j\omega})| \equiv 1 \text{ for all } \omega \tag{7}$$

$$\arg\{H_{id}(e^{j\omega})\} = \Theta_{id}(\omega) = -D\omega \tag{8}$$

respectively. The phase information is often represented in the form of *group delay*, defined as the negative frequency derivative of the phase

$$\tau_g(\omega) = -\frac{\partial\Theta(\omega)}{\partial\omega} \tag{9a}$$

or via *phase delay*

$$\tau_p(\omega) = -\frac{\Theta(\omega)}{\omega} \tag{9b}$$

(see [81] or [84]).

$$x(n) \longrightarrow \boxed{z^{-D}} \longrightarrow y(n) = x(n-D)$$

2. An ideal discrete-time delay system.

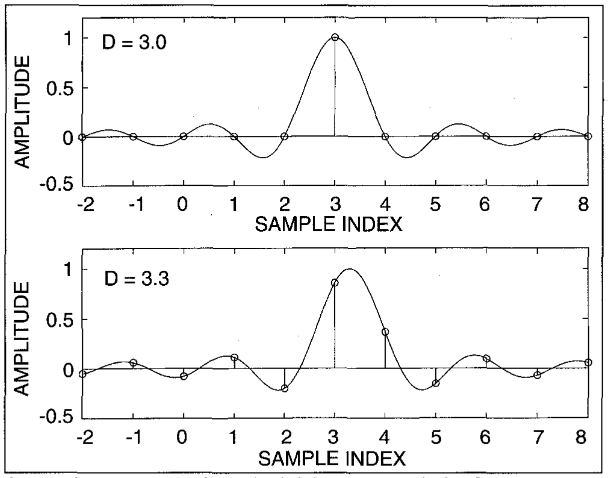
Both the group delay and the phase delay are measures for the delay of the system. Their difference is typically illustrated by considering an amplitude modulated signal for which the phase delay tells the delay of the carrier signal and the group delay tells that of the envelope (baseband signal) [81]. When the phase is exactly linear, the two delay measures yield identical results. For the frequency response Eq. 6, both the group delay and the phase delay are equal to the constant value D in the whole frequency band, i.e.,

$$\tau_{p,id}(\omega) = D \tag{10a}$$

$$\tau_{g,id}(\omega) = D \tag{10b}$$

Which of the three delay measures—phase, group delay, or phase delay—should be used? The physical viewpoint suggests the use of the phase or phase delay when the delaying of individual sinusoidal components of the signal is of interest. The group delay and the phase delay give flat curves for linear or almost-linear systems, which is more illustrative than the often steep-slope phase curve. As we are primarily interested in the delay of sinusoidal components, we have chosen to use the phase delay in the plots of this article. However, when introducing approximation methods, the phase or the group delay are also employed.

To summarize our discussion, we can use the ideal z-domain transfer function Eq. 4 or the ideal frequency response Eq. 6 as the design objective. These functions are complex-valued so that both the real and imaginary parts must be taken into account, separately or together. Alternatively, we can express this information in terms of magnitude and phase where the phase can be replaced by the group delay or the phase delay.



3. Impulse response of an ideal delay filter with the delay a) D = 3 and b) D = 3.3.

Ideal Solution

Assuming that the (real-valued) discrete-time signal represents a bandlimited baseband signal, the implementation of a constant delay can be considered as an approximation of the ideal discrete-time linear-phase allpass filter with unity magnitude and constant group delay of the given value D. The corresponding impulse response is obtained via the inverse discrete-time Fourier transform [84]:

$$h(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} H(e^{j\omega}) e^{j\omega n} d\omega \text{ for all } n \tag{11}$$

Substitution of Eq. 6 into Eq. 11 yields the solution for the ideal impulse response as

$$h_{id}(n) = \frac{\sin[\pi(n-D)]}{\pi(n-D)} = \text{sinc}(n-D) \text{ for all } n \tag{12}$$

which has the shape of the familiar sinc function defined as

$$\text{sinc}(x) = \frac{\sin(\pi x)}{\pi x} \tag{13}$$

When the desired delay D assumes an integer value, the impulse response Eq. 12 reduces to a single impulse at $n = D$, but for noninteger values of D the impulse response is an infinitely long, shifted and sampled version of the sinc function (Fig. 3). Unfortunately, the ideal impulse response is not only infinitely long but also noncausal, which makes it impossible to implement it in real-time applications.

Equation 12 gives an answer to the original problem, i.e., where the delayed signal value should be placed as it cannot be put "between the samples." In the ideal case, it is to be spread over all the discrete-time signal values, weighted by appropriate values of the sinc function.

This simple result (Eq. 12) is of fundamental importance since, whatever method is used, the impulse response of the approximating (real-coefficient) filter must imitate this ideal response in some meaningful sense. It is also evident that with a finite-order causal FIR or IIR filter the ideal response can only be approximated. Furthermore, the ideal solution from Eq. 12 can be utilized in the formulation of the approximation problem in the time domain, as will be discussed in more detail in the following sections.

Fractional Delay Approximation Using FIR Filters

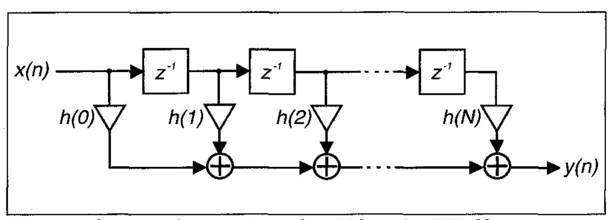
Let us consider the approximation of the ideal fractional delay D by an Nth-order (length $L = N + 1$) FIR filter with the z-domain transfer function

$$H(z) = \sum_{n=0}^{N} h(n) z^{-n} \tag{14}$$

An FIR filter is typically implemented with the direct form structure of Fig. 4. It is now desired to determine the coefficients $h(n)$ such that the chosen norm of the frequency-domain error function

$$E(e^{j\omega}) = H(e^{j\omega}) - H_{id}(e^{j\omega}) \tag{15}$$

is minimized. Note that both the desired function $H_{id}(e^{j\omega})$ and the error function are *complex-valued*. This complicates the solution in general as compared to a real-valued approximation problem in linear-phase FIR filter design [84]. However, simple solutions can still be found as will be shown in the following example.



4. Direct form implementation of an Nth-order FIR filter.

Application 1: Synchronization in Digital Modems

In digital data transmission, the main task of the receiver is to detect the transmitted data symbols as reliably as possible. To this end, the receiver must be synchronized to the symbols of the incoming data signal. Since the frequency of an analog oscillator tends to vary with time and temperature, and in mobile communications Doppler frequency shifts must also be accounted for, it is essential that the synchronization is monitored during all the transmission.

A traditional solution for symbol synchronization has been to use 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. An example of this kind of a receiver structure is illustrated in Fig. A [32].

Since the postprocessing of the sampled data signal (matched filtering and data detection) is performed digitally anyway, it is often advantageous to implement the synchronization using digital techniques as well. A digital solution is outlined in Fig. B. The local oscillator controlling the sampling is now autonomous, and all the timing is fine-tuned in the digital domain by controlling an appropriate FD filter. For this application, low-order FD filters are usually sufficient. A third-order FIR filter implemented with the Farrow structure has been proposed by Erup, et al. [27]. For excellent overviews of FD filter design for synchronization applications, see [32], [27], and [72].

An alternative to a separate FD filter for synchronization is to embed the delay control in another filtering function, i.e., in a matched filter. This approach has been studied in [95] and [96]. The application of FD filters or interpolators to synchronization of modems has also been considered in [29], [8], [30], [31], [66], [67], [33], [7], and [10].

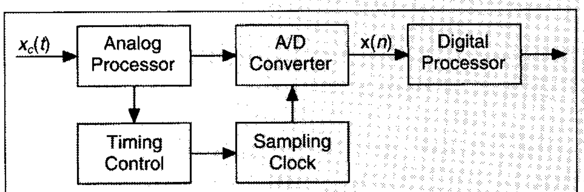


Fig. A. Analog synchronization via controlling the sampling time (after [32]).

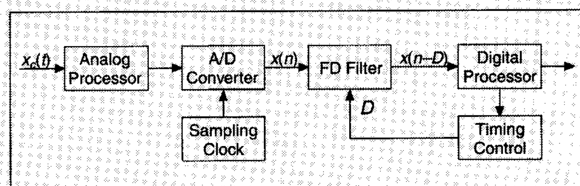


Fig. B. Digital synchronization via an FD filter (adapted from [32]).

Least Squared Error Solutions for FIR Filters

(1) Direct Least Squared Integral Error FIR Design

A mathematically straightforward solution to the fractional delay problem is obtained by a least squared error (LS) type design. The L_2 norm [84] of the error function can be defined as

$$E_1 = \frac{1}{\pi}\int_0^{\pi} |E(e^{j\omega})|^2 \, d\omega$$
$$= \frac{1}{\pi}\int_0^{\pi} |H(e^{j\omega}) - H_{id}(e^{j\omega})|^2 \, d\omega \qquad (16)$$

(The subscript 1 in E_1 denotes here the particular error measure, not the chosen L_p norm.)

Via the Parseval relation this frequency-domain error measure can be converted into the time domain, resulting in the alternative expression

$$E_1 = \sum_{n=-\infty}^{\infty} |h(n) - h_{id}(n)|^2 \qquad (17)$$

which facilitates the use of the ideal solution for $h_{id}(n)$ derived in the previous section. It is readily seen that, as is the case with linear-phase FIR filter design, the L_2-optimal Nth-order FIR filter is obtained by simply truncating the ideal impulse response to $L = N + 1$ terms [84]. The optimal causal solution can be expressed as

$$h(n) = \begin{cases} \operatorname{sinc}(n-D) & \text{for } M \le n \le M+N \\ 0 & \text{otherwise} \end{cases} \qquad (18)$$

where M is the (integer) time index of the first nonzero value of the impulse response. For causality, it must be assumed M 0. If the desired delay D is very large, it is practical to realize part of it as a chain of M unit delays and the rest with an FD filter. The solution Eq. 18 can be modified for approximation in the general lowpass interval $(0, \alpha\pi)$, which yields the impulse response

$$h(n) = \begin{cases} \alpha \operatorname{sinc}[\alpha(n-D)] & \text{for } M \le n \le M+N \\ 0 & \text{otherwise} \end{cases} \qquad (19)$$

The resulting approximation error of Eq. 18 or Eq. 19 can be expressed in the time domain as:

Application 2: Incommensurate Sampling Rate Conversion

Changing the sampling rate of a digital signal is straightforward as long as the ratio of the input and output sampling rates is an integer or a ratio of small integers (see [24], [119], and [6]). This kind of sampling rate conversion is implemented efficiently with a polyphase structure, which can be considered as a periodically time-varying filter. Each polyphase branch is a fractional delay filter approximating an integer number of fractions of the input sample interval.

The sampling rate conversion for incommensurate (irrational) ratios is more difficult. A conceptually simple solution is obtained using fractional delay filters. Figure A illustrates two sampling grids where the locations of input and the desired output samples are shown on the time axis. It is obvious that every output sample can be obtained from the input samples by applying an FD filter approximating an appropriate delay d_k. The essential tasks are thus to implement 1) a control unit that computes the needed delay values and 2) a computationally efficient FD filter whose coefficients change for every output sample. A block diagram of an incommensurate-ratio sampling rate conversion system is presented in Fig. B.

A multistage FIR interpolator structure was used in [57] for conversion ratios which involve large integer numbers. Ramstad [91], [92] has considered several approaches, such as a hybrid technique combining a high-order FIR interpolator with a low-order Lagrange interpolator. The same principle was proposed in [103] where an efficient table look-up technique with linear interpolation for FD filter coefficient update was employed. The two-stage approach has been extended using a spline interpolator as an FD filter in the second stage [25]. In [112], a high-quality fixed-coefficient IIR interpolator was employed as a first stage to reduce the complexity of the time-varying FIR FD filter. The use of spline interpolator for sampling rate conversion was proposed in [137]. A Lagrange interpolator realized with the Farrow structure was considered in [55]. Other references on the topic include [3], [82], and [86].

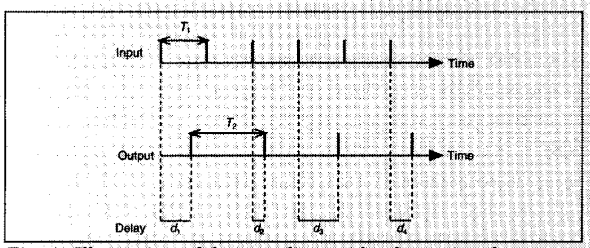


Fig. A. Illustration of the sampling grids of two signals sampled at rates f_{s1} and f_{s2}, respectively, and the fractional delays, d_k, between the contiguous samples of these signals.

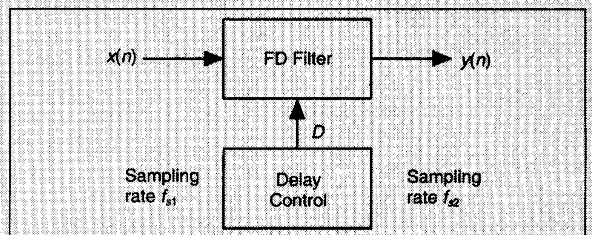


Fig. B. System for incommensurate-ratio sampling rate conversion.

$$E_1 = \sum_{n=-\infty}^{M-1} |h_{id}(n)|^2 + \sum_{n=M+N+1}^{\infty} |h_{id}(n)|^2 \qquad (20)$$

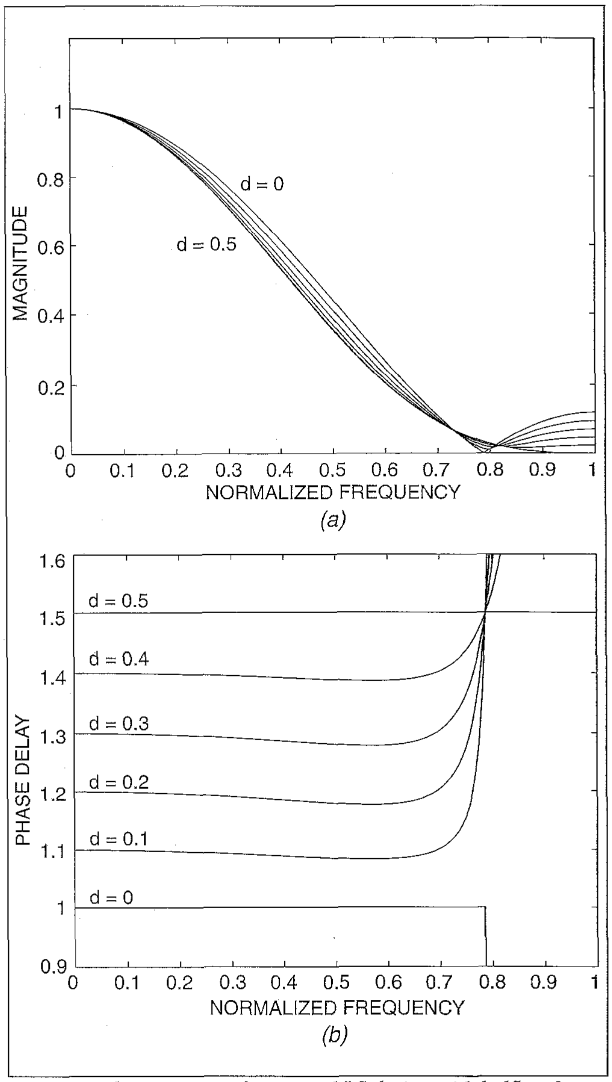
which leads to two important observations. First, the approximation error clearly decreases as the filter order increases. Second, the smallest error for a given filter order is obtained when the overall delay D is placed at the 'center of gravity' of the ideal impulse response [81], i.e.,

$$M_{opt} = \begin{cases} \text{Round } (D) - \dfrac{N}{2} & \text{for even } N \\ \text{Int } (D) - \dfrac{N-1}{2} & \text{for odd } N \end{cases} \qquad (21)$$

where Round $(\cdot)$ denotes rounding to the nearest integer. This is due to the fact that the values of the sinc function are largest in magnitude around the center value ($n = D$) which should thus be placed in the middle of the truncated impulse response. The rule Eq. 21 generally applies for other approximation criteria as well. In [46], a related stochastic case was considered and it was shown that linear interpolation from N samples of an autoregressive process yields a minimum least squared error when interpolating the midpoint value of the data record.

Moreover, it is observed that for a nonzero M a part of the total delay D is implemented by a cascade of M unit delay elements. Clearly, the total delay can be made arbitrarily long. However, it cannot be made arbitrarily short without affecting the precision of approximation which depends on the filter length $L = N + 1$. In other words, the desired total delay D sets an upper bound for the precision of approximation, whatever approximation criterion is used.

To simplify notation, we set $M = 0$ for the rest of the article and assume that the filter order is chosen such that Eq. 21 holds. Additional delays can always be included in the system when needed. Although optimal in the L_2 sense, the truncated impulse response (Eq. 18) has a well-known feature, the Gibbs phenomenon, which causes ripple in the magnitude response. This is usually not desired. In fullband approximation, the Gibbs phenomenon means deterioration of the



5. Magnitude responses of truncated LS design with halfband approximation (α = 0.5).

magnitude response close to the Nyquist frequency ($\omega = \pi$). The peak value of this ripple is approximately constant irrespective of the filter order (for high filter orders). Also the phase delay response suffers from similar characteristics: when the filter order is increased, the peak phase delay error remains approximately constant.

The magnitude and phase delay responses are illustrated in Fig. A1 (see Appendix A). Only the responses corresponding to the delay values d = 0.0, 0.1, ..., 0.5 are shown; the magnitude responses are symmetrically equivalent, e.g., that of d = 0.1 is equal to that of d = 0.9 for even-length filters, and the corresponding phase delay responses are symmetric with respect to the curve of d = 0.5.

For comparison, Fig. 5 illustrates the magnitude and phase responses of a length-4 half-band fractional delay filter designed using Eq. 19 with α = 0.5. It is seen that the passband is now very narrow, so that the bandwidth reduction alone does not provide a good solution for all purposes.

(2) Windowing Methods for FIR Filter Design

The performance of a fractional delay filter obtained by truncating the sinc solution is usually not acceptable in practice. A well-known method to reduce the Gibbs phenomenon is to use window functions for time-domain weighting. Instead of truncating the impulse response, a bell-shaped window can be used which puts more emphasis on the middle values of the impulse response and reduces the peak magnitude error at the cost of a wider transition band of the filter. The new windowed impulse response is obtained as

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

where the ideal impulse response $h_{id}(n)$ is truncated and shaped by multiplying by $w(n\text{-}D)$, which is a length-L ($= N +$ 1) window sequence shifted by the appropriate delay value, D. Many continuous window functions, e.g., the Hamming window, can be easily delayed by a fractional value [14, 16, 17]. Note that the obtained impulse response does not directly minimize any least squared error measure, but it is a modification of the least squares solution minimizing E_1. Hence, the windowed version inevitably has larger least squared error than the original truncated impulse response.

A comprehensive review of window functions was presented in [35]. The Kaiser window allows the control of the peak ripple using one parameter. The Dolph-Chebyshev window is optimal in the sense that it has the smallest possible peak sidelobe ripple, which is given as a parameter. It is easy to use since it is included in certain commercial mathematical program packages [69]. Figure A2 shows the responses of the filters in Fig. A1 when 40 dB Dolph-Chebyshev windows are used, with a clearly lower ripple but also with a wider transition band. We used fractionally shifted windows employing the approximate shifting technique introduced in [56].

In principle, window-based design is fast and easy. However, in practice it is somewhat difficult to control the magnitude error by adjusting window parameters—this is particularly true for very short filter lengths (L less than 10), which are typical in many applications. However, if the detailed error behavior is not critical, the method is quite suitable for real-time coefficient update. One can either store the window in memory and compute the values of the sinc function on-line or, for example, store enough samples of the windowed sinc function in the memory and compute the filter coefficients using interpolation [103, 106].

(3) FIR Filter Design with Smooth Transition Band Functions

The window functions effectively reduce the approximation bandwidth of the fractional delay filter in order to obtain a better approximation in the reduced band. This can also be achieved by modifying the desired magnitude response. It is, however, essential that the desired response is smooth: the Gibbs phenomenon is caused by the discontinuity and that is why reducing the bandwidth as in Eq. 19 does not help.

Application 3: Fractional Delay Improves Speech Coding

Linear predictive speech coding is based on two main processing steps: modeling of the vocal tract resonator system and modeling of the acoustic source or excitation. In linear prediction, new samples are predicted by forming a linear (weighted) combination of the past p samples. Since the shape of the vocal tract and, accordingly, its resonances are varying relatively slowly, the weights, called predictor coefficients, have to be determined only every 20-40 ms.

The predictor coefficients and the corresponding vocal tract filter are solved based on short term prediction, where only 8-16 past samples are used to predict the new one. When the filter part is determined, the source signal is solved by inverse filtering of the speech signal. The output of the inverse filter ($e(n)$ in Fig. A) is called prediction error, or the residual. In the case of a voiced speech segment such as a vowel, the error signal resembles a pulse train. The time period between the pulses determines the fundamental frequency of the sound. This is perceived as the pitch of a voice.

From the coding point of view, it is practical to estimate one parameter that determines the fundamental frequency of the pulsed excitation. A long term predictor (pitch predictor) is used to estimate the pitch period (e.g., see [89]). Conventionally, the pitch predictor consists of a delay line, which approximates the pitch period and, of an additional three tap FIR filter to improve the prediction. Together, four parameters have to be encoded: the length of the delay line and the three predictor coefficients.

A typical sampling frequency of a speech coder is 8 kHz. Considering female speech with a fundamental frequency of 250 Hz and pitch period of 4 ms (32 samples), an error of one sample will introduce a random variation of about 3 percent in the fundamental frequency estimate, leading to clear roughness in the voice quality.

A fractional delay pitch predictor has been proposed by Kroon and Atal [50, 51]. In this structure, only two parameters have to be encoded: the length of the delay line (with integer and fractional parts), and one predictor coefficient (see Fig. B). The results show that with male voices, this predictor gives about the same predictor gain (SNR) as the conventional one, however, now with fewer bits. Additionally, the gain and quality was even improved with female voices, as one can predict from the example given above. The application of fractional delay filters to pitch prediction has also been addressed in [68, 70, 78].

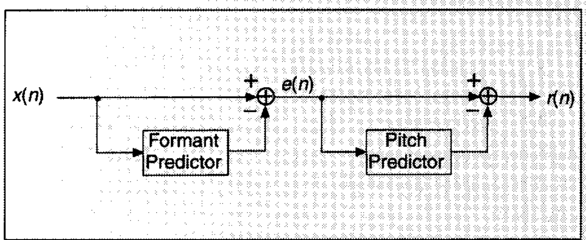


Fig A. Block diagram of the analysis part of a speech coder with a formant and pitch predictor in cascade (after [89]).

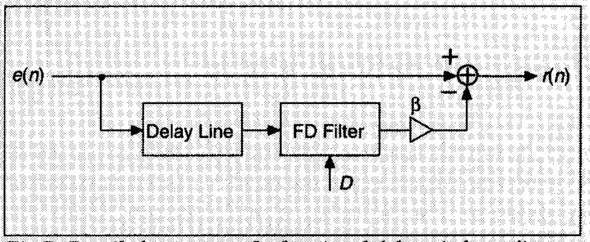


Fig B. Detailed structure of a fractional delay pitch predictor.

A way to define smooth lowpass responses for linear-phase FIR design was presented in [84] and [12]. When the desired frequency response is defined to be a smooth function instead of the discontinuous rectangle, the impulse response decays much faster and may be truncated without explicit windowing. For simple analytic transition functions, like splines or trigonometric functions, the impulse response can be solved in closed form. This technique can also be applied to the fractional delay filter design.

Instead of defining the desired frequency response as in Eq. 6, we can prescribe a passband $[0, \omega_p]$ with the ideal response $e^{-jD\omega}$, a stopband $[\omega_s, \pi]$ with ideal response 0, and a Pth-order spline (times $e^{-jD\omega}$) in the transition band. This results in the following impulse response

$$h(n) = \left\{ \frac{\sin\left[\frac{n-D}{2P}(\omega_s - \omega_p)\right]}{\frac{n-D}{2P}(\omega_s - \omega_p)} \right\}^P \frac{\sin\left[\frac{n-D}{2}(\omega_s + \omega_p)\right]}{\frac{\pi(n-D)}{2P}} \tag{23}$$

for $n = 0,1,2,\dots,N$.

For a detailed derivation of lowpass filter design, see [84]. Note that the used error norm is the same as in direct LS design (Eq. 16), but the desired frequency response is different.

This approach offers a simple way to control the approximation band and yet is suitable for on-line coefficient calculations. A design example is presented in Fig. A3. A remarkable feature is that the magnitude response remains constant with high precision when the fractional delay is changed, unlike with previous designs. However, this comes at the cost of a worse phase delay response, which is particularly severe with the 4-tap example (see Fig. A3b).

Other transition functions are not considered here. The impulse response of a raised-cosine transition function is easily obtained by modifying the result of [12].

4) General Least Squares FIR Approximation of a Complex Frequency Response

The smooth-transition-band-method reduces the error of the fractional delay filter in an advantageous way, but it has some shortcomings. First, the impulse response is still infinitely long although it decays fast, and the truncation always causes some error. Second, the desired response must be defined explicitly in the whole Nyquist band. While the method has the clear design advantage that the inverse Fourier transform can be used to determine the coefficients explicitly, it nevertheless forces approximation resources to be wasted outside the approximation band where they are actually not needed.

In principle, the 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. This scheme also enables frequency-domain weighting of the LS error; it results in the following error formulation (alternative formulations employing eigenfilter techniques have been introduced in [76] and [85]).

$$E_4 = \frac{1}{\pi}\int_0^{\alpha\pi} W(\omega)|E(e^{j\omega})|^2\, d\omega = \frac{1}{\pi}\int_0^{\alpha\pi} W(\omega)|H(e^{j\omega}) - H_{id}(e^{j\omega})|^2\, d\omega \tag{24}$$

where the error is defined in the lowpass frequency band $[0, \alpha\pi]$ only and $W(\omega)$ is the nonnegative frequency-domain weighting function. Note that $W(\omega)$ has nothing to do with the time-domain window $w(n)$ (Eq. 22).

We now derive the solution for a general $H_{id}(e^{j\omega})$. To formulate the solution in compact form, let us introduce vector notation as

$$\mathbf{h} = [h(0) \quad h(1) \quad h(N)]^T \tag{25a}$$

$$\mathbf{e} = [\,1 \quad e^{-j\omega} \ldots e^{-jN\omega}]^T \tag{25b}$$

for the filter coefficients and for the discrete-time Fourier transform, respectively, and the matrix

$$\mathbf{C} = \mathrm{Re}\{\mathbf{e}\mathbf{e}^H\} = \begin{bmatrix} 1 & \cos\omega & \ldots & \cos(N\omega) \\ \cos(\omega) & 1 & \ldots & \cos[N-1]\omega \\ \vdots & & \ddots & \vdots \\ \cos(N\omega) & \cos[(N-1)\omega] & \ldots & 1 \end{bmatrix} \tag{26}$$

where the superscript 'H' stands for the Hermitian operation, i.e., transposition with conjugation. Now we can express the Fourier transform as $H(e^{j\omega})$ and the error measure Eq. 24 becomes

$$E_4 = \frac{1}{\pi}\int_0^{\alpha\pi} W(\omega)[\mathbf{h}^T\mathbf{e} - H_{id}(e^{j\omega})][\mathbf{h}^T\mathbf{e} - H_{id}(e^{j\omega})]^*\, d\omega = \frac{1}{\pi}\int_0^{\alpha\pi} W(\omega)[\mathbf{h}^T\mathbf{C}\mathbf{h} - 2\mathbf{h}^T\,\mathrm{Re}\{H_{id}(e^{j\omega})\mathbf{e}^*\} + |H_{id}(e^{j\omega})|^2\,]d\omega \tag{27}$$

where the superscript '*' denotes complex conjugation. This can further be put into the following form

$$E_4 = \mathbf{h}^T\mathbf{P}\mathbf{h} - 2\mathbf{h}^T\mathbf{p}_1 + p_0 \tag{28}$$

where

$$\mathbf{P} = \frac{1}{\pi}\int_0^{\alpha\pi} W(\omega)\mathbf{C}\, d\omega \tag{29a}$$

$$\mathbf{p}_1 = \frac{1}{\pi}\int_0^{\alpha\pi} W(\omega)\Big[\mathrm{Re}\{H_{id}(e^{j\omega})\}\mathbf{c} - \mathrm{Im}\{H_{id}(e^{j\omega})\}\mathbf{s}\Big]d\omega \tag{29b}$$

$$p_0 = \frac{1}{\pi}\int_0^{\alpha\pi} W(\omega)|H_{id}(e^{j\omega})|^2\, d\omega \tag{29c}$$

$$\mathbf{c} = [1 \quad \cos(\omega) \quad \ldots \quad \cos(N\omega)]^T \tag{30a}$$

$$\mathbf{s} = [0 \quad \sin(\omega) \quad \ldots \quad \sin(N\omega)]^T \tag{30b}$$

The error measure (Eq. 28) is quadratic with a unique minimum-error solution that is found by setting its derivative with respect to $\mathbf{h}$ to zero. This results in the following normal equation

$$2\mathbf{P}\mathbf{h} = 2\mathbf{p}_1 = 0 \tag{31}$$

which is solved formally by matrix inversion, i.e.,

$$\mathbf{h} = \mathbf{P}^{-1}\mathbf{p}_1 \tag{32}$$

Hence, the optimal solution is obtained by determining the integrals involved in Eqs. 29 (usually numerically) and solving the set of $(N+1)$ linear equations (Eq. 32). The arithmetic complexity (i.e., the required number of multiplications and additions) of a matrix inversion is in general proportional to $(N+1)^3$ so that the computational costs are considerably increased over the previous methods, where the coefficients are obtained in explicit form. Furthermore, numerical problems may arise, particularly in narrowband approximation [60, 13]. This is the price to be paid for the increased flexibility of the design.

The numerical problems are indeed regrettable as the general least squares approximation of a complex response

Application 4: Tuning to Music

Physical modeling of musical instruments is an emerging approach to sound synthesis. The idea is not merely to imitate the sound of a given instrument but to design a computational model of its sound producing mechanism to guarantee natural sound quality in different playing situations.

The Karplus-Strong model [45] was among the first to show that musical instruments can be modeled and synthesized in real time using DSP techniques. Many fundamental problems were solved by Jaffe and Smith [39] to make this approach more practical. They also demonstrated that fractional delay filters were needed for successful synthesis.

Figure A shows a signal processing model for a vibrating string. The pitch of the sound is controlled by the length of the delay line in the feedback loop. It is implemented as a combination of an integer-length delay line and a fractional delay filter. In addition, a lowpass filter (loop filter) is needed to simulate physical losses in the string. The input signal x(n) is a short pulse that approximates the plucking sound.

Figure B illustrates the limitations of using an integer-length delay line. The difference between the desired pitch values of a string model and the quantized values due to an integer-length delay line are shown. It can readily be noticed that the pitch values available without fractional delays do not match with any generally used musical scales at high frequencies. In [39], a first-order all-pass FD filter (equivalent to the first-order Thiran allpass filter) was employed for fine-tuning the delay-line length. Karjalainen and Laine [42] proposed the use of a third-order Lagrange FD filter. The synthesis of string instruments using fractional delay filters has also been discussed in [110], [43], [44], [129], and [133].

Similarly, models of wind instruments can be built upon variable-length delay lines (see, e.g., [21], [22], [120], [121]). Fractional delay waveguide filters (FDWF), as defined in [122], [124], [125], and [126], are flexible extensions to the fractional delay theory. They allow for modeling of wave scattering junctions such as finger holes [123] and conical tube sections in wind instruments [127] as well as variable-length sections of the vocal tract in articulatory speech synthesis [125], [124], [128]. A survey of FDWF's is given in [126] and [135].

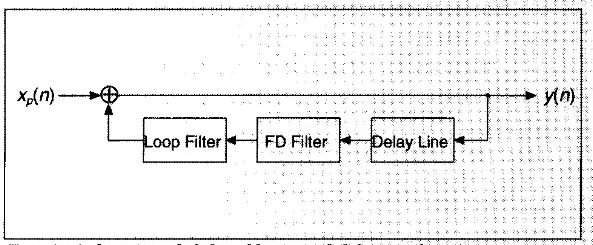


Fig. A. A fractional delay filter model for a vibrating string.

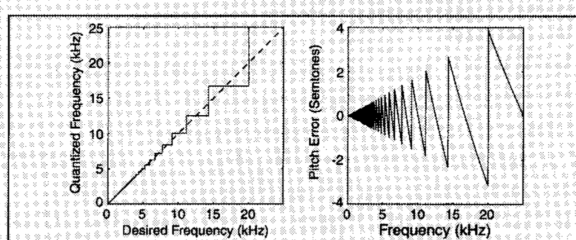


Fig. B. (a) Desired vs. actual fundamental frequency of a string model with integer-length loop delays for a 50 kHz sampling rate, and (b) corresponding approximation error in semitones (after [39]).

would be an extremely versatile and easy-to-use FIR design technique for any application. As discussed in [13] in the context of linear-phase multiband FIR filter design, the numerical problems (the ill-conditioning of matrix **P**) are caused exactly by the advantageous "don't care" bands where no response is specified, and these problems can be alleviated by specifying a response there. In [13], optimized spline functions were employed to define transition band responses to reduce the ill-conditioning. However, this may result in a solution biased from the optimal one if the transition response is not carefully selected.

When employing the desired function $H_{id}(e^{j\omega}) = e^{-jD\omega}$ of Eq. 6, the integrals (Eq. 29) are greatly simplified. Particularly, if we choose $W(\omega) = 1$, they can be given in closed form. The elements of **P** and $\mathbf{p}_1$ can be expressed as

$$P_{k,l} = \frac{1}{\pi}\int_0^{\alpha\pi} \cos[(k-l)\omega]d\omega = \alpha\,\mathrm{sinc}[\alpha(k-l)] \qquad k,l = 1,2,\ldots,L \tag{33a}$$

$$p_{1,k} = \frac{1}{\pi}\int_0^{\alpha\pi} \cos[(k-D)\omega]d\omega = \alpha\,\mathrm{sinc}[\alpha(k-D)] \qquad k = 1,2,\ldots,L \tag{33b}$$

It is, however, worth noting that the **P** matrix is independent of the delay value D and only needs to be inverted once. Furthermore, the Toeplitz structure of the **P** matrix enables the use of the fast Levinson algorithm [36] in its inversion.

Obviously, the resulting fractional delay filter depends on the choice for the frequency band of approximation and on the weight function. As a simple example, we illustrate here the design of a fractional delay filter with the approximation band $[0, \pi/2]$ with unity weight function $W(\omega) = 1$ in Fig. A4. Note that this is not equivalent to the case previously where we prescribed the same passband but could not define a "don't care" band. It is seen that the peak magnitude error is greatly reduced as desired. This is achieved at the cost of a slightly increased error outside the approximation band.

5) Least Squares FIR Design on a Discrete Frequency Grid
The above methods can be modified for the design on a discrete (uniformly or nonuniformly spaced) frequency grid. For example, the inverse discrete-time Fourier transform (IDTFT) employed in Eqs. 11 and 23 can be replaced with the inverse discrete Fourier transform (IDFT) where the frequency variable is also discrete-valued. Furthermore, if the integration in Eq. 29 is carried out using the trapezoidal rule, it effectively results in the discretization of the frequency variable. As these versions bring little new to the above formulations, we omit a detailed treatment here. However, computations on a discrete frequency grid in general result in numerically more robust algorithms [64], [34]. In [84] such methods are discussed in the context of linear-phase FIR filter design.

6) Stochastic Least-Mean-Squared (LMS) Error FIR Interpolation
Above, we have explicitly utilized only knowledge of the desired frequency response in the design of the fractional delay filter. The character of the signal can only be utilized in a coarse manner, e.g., by specifying the approximation band according to the energy distribution of the signal.

However, more knowledge about the signal can be utilized to achieve small errors. Signals are often stochastic (random) in nature and characterized by their average power spectrum or, equivalently, by the autocorrelation function. This stochastic approach was introduced for upsampling interpolation in [79]. For our case, the criterion to be minimized can be formulated as the minimum expected mean squared output error, or

$$\begin{aligned} E_6 &= E\{|y(n) - y_{id}(n)|^2\} = E\{|x(n) * [h(n) - h_{id}(n)]|^2\} \\ &= \frac{1}{\pi}\int_0^{\pi} S_{xx}(\omega)|H(e^{j\omega}) - H_{id}(e^{j\omega})|^2\, d\omega \end{aligned} \tag{34}$$

where $x(n)$ and $y(n)$ are the input and the output signals of the delay filter, $E\{\}$ denotes the expectation value of the argument, $H_{id}(e^{j\omega}) = e^{-jD\omega}$ is the ideal frequency response, and $*$ denotes convolution. $S_{xx}(\omega)$ is the average power spectrum of the input signal $x(n)$.

The error E_6 of Eq. 34 is formally equivalent to the general complex least squared error E_4 of Eq. 24 when the weight function is chosen as

$$W(\omega) = S_{xx}(\omega) \tag{35}$$

Hence, the formal solution is obtained directly by matrix inversion as in Eq. 32, with the given modifications. Note that in this case the elements of matrix $\mathbf{P}$ are values of the autocorrelation function of the input signal $x(n)$, i.e.,

$$\begin{aligned} P_{k,l} &= \frac{1}{\pi}\int_0^{\pi} S_{xx}(\omega)\cos[(k-l)\omega]\, d\omega \\ &= E\{x(n)x(n+k+1)\} = r_{xx}(k-l) \end{aligned} \tag{36}$$

Provided that the input power spectrum $S_{xx}(\omega)$ can be estimated beforehand, this scheme is also suitable for real-time update since the $\mathbf{P}$ matrix only needs to be inverted once.

Maximally Flat FIR FD Filter Design: Lagrange Interpolation

Instead of minimizing a least squared error measure, the error function can be made maximally flat at a certain frequency, typically at $\omega_0 = 0$, so that the approximation is at its best close to this frequency. This means that the derivatives of the frequency-domain error function are set to zero at this point, that is

$$\left.\frac{d^n E(e^{j\omega})}{d\omega^n}\right|_{\omega=\omega_0} = 0 \quad \text{for } n = 0,1,2,\ldots N \tag{37}$$

where $E(e^{j\omega})$ is the complex error function (Eq. 15) with the desired response $H_{id}(e^{j}\omega) = e^{-jD}\omega$ as in Eq. 6. Differentiating and inserting the value $\omega_0 = 0$ in Eq. 37, this set of $L = N + 1$ linear equations can be expressed in terms of the impulse response as

$$\sum_{k=0}^{N} k^n h(k) = D^n \quad \text{for } n = 0,1,2,\ldots N \tag{38}$$

or, in matrix notation

$$\mathbf{Vh} = \mathbf{V} \tag{39}$$

where $\mathbf{h}$ is the coefficient vector (25a) and

$$\mathbf{V} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 0 & 1 & 2 & & N \\ 0 & 1 & 2^2 & & N^2 \\ \vdots & & & \ddots & \vdots \\ 0 & 1 & 2^N & \cdots & N^N \end{bmatrix} \tag{40a}$$

is an $L \times L$ *Vandermonde* matrix [80] and

$$\mathbf{v} = [1 \;\; D \;\; D^2 \;\; \ldots \;\; D^N]^T \tag{40b}$$

As discussed by Oetken [80], the solution to Eq. 39, is equal to the classical *Lagrange interpolation* formula, where the coefficients are obtained by fitting the interpolating polynomial to pass through a given set of data values. For the z-transform, this means that

Design Guide 1: Simplest FIR FD Filter—Lagrange Interpolator

Lagrange interpolation is probably the easiest way to design an FIR filter to approximate a given fractional delay D. The coefficients are obtained from Eq. 42, that is

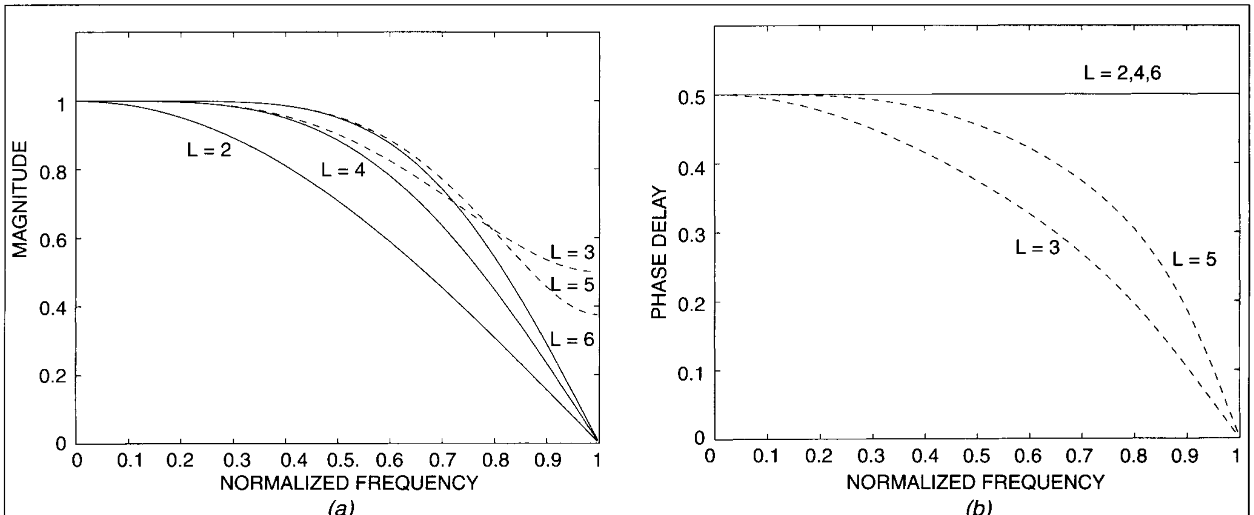
$$h(n)=\prod_{\substack{k=0\\k\neq n}}^{N}\frac{D-k}{n-k}\ \text{for}\ n=0,1,2,\ldots N$$

where N is the order of the filter. The coefficients for the Lagrange FD filters of order N = 1, 2, and 3 (or, equivalently, lengths L = 2, 3, and 4) are given in the Table.

The plot of the magnitude and phase delay responses of the Lagrange interpolators of order N = 1 to 5 (L = 2 to 6) in the worst-case approximation (half-sample delay) are shown in Fig. 6. It is seen that these low-order filters give an excellent approximation at low frequencies. However, the approximation bandwidth grows very slowly when the filter order is increased. Thus, if a 4-tap Lagrange FD filter is not good enough for a given purpose, it may be better to use an LS-based FIR filter design method instead.

Table: Coefficients of the Lagrange FD Filters of Order N= 1, 2, and 3

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



6. a) Magnitude and b) phase delay responses of Lagrange interpolating filters of length L = 2, 3, 4, 5, and 6 with d = 0.5.

$$H(z)=z^{-D}\quad \text{for } D=0,1,2,\ldots N \tag{41}$$

or that for integer values of the desired delay the approximation error is set to zero. The solution can be given in an explicit form as

$$h(n)=\prod_{\substack{k=0\\k\neq n}}^{N}\frac{D-k}{n-k}\quad for\ n=0,1,2,\ldots N \tag{42}$$

The case N = 1 corresponds to *linear interpolation* between two samples. In this case the two coefficients are

$$h(0)=1-D,\ h(1)=D \tag{43}$$

Naturally, here the integer part of D is zero so that $D = d$. The amplitude and phase delay responses of low-order Lagrange interpolators are shown in Fig. 6 for d = 0.5 and N = 1, 2, ..., 5 (L = 2, 3, ..., 6; only the fractional part of the phase delay is shown). It is seen that, due to the coefficient symmetry for d = 0.5, the even-length filters (L = 2, 4, and 6) are exactly linear-phase, but the magnitude responses suffer from the zero at $\omega=\pi$. The odd-length filters (L = 3 and 5) have better magnitude responses, but the phase delays are worse. This magnitude-phase delay tradeoff between even and odd-length filters is similar for other fractional delay values and for other approximation methods as well.

Lagrange interpolation has several advantages: easy explicit formulas for the coefficients, very good response at low frequencies, and a smooth magnitude response. The maximum of the magnitude response never exceeds unity when the delay is near to half the filter length. This is important in applications including feedback. On the other hand, the approximation error is often unnecessarily small at low frequencies, at the cost of the performance at higher frequencies. Examples of Lagrange interpolator design with different values of d are shown in Fig. A5.

Lagrange interpolation has been proposed for the approximation of a fractional delay independently by Laine [58] and by Liu and Wei [66, 67]. In the context of multirate filters, the Lagrange interpolation scheme has been known for a long time [24, 80, 97]. Ko and Lim [48] derived a general maximally flat frequency-error solution at an arbitrary frequency $\omega=\omega_0$, which with the choice ω_0 reduces to Lagrange interpolation. The explicit (complex-valued) solution for the case of nonzero ω_0 was presented by Hermanowicz [37]. The maximally flat FIR FD design has also been discussed by Sivanand *et al.* [101, 102].

In [73], the maximally flat interpolator design was achieved by truncating the Taylor series of the error function (15) and by forcing the derivatives in the Taylor series to be zero at $\omega=\omega_0$. In [49] it was shown that the Lagrange solution Eq. 42 can also be obtained from the ideal sinc solution Eq. 18 using the windowing method. The window coefficients are computed using the binomial formula.

Also other polynomial interpolation techniques, such as splines [137], have been suggested for fractional delay approximation or interpolation. These techniques are nonoptimal from the frequency-domain viewpoint. Still, for applications where good accuracy at high frequencies is not required, for example the parabolic interpolation technique [27] may be attractive since it can be implemented efficiently with a third-order FIR filter.

One can also construct a mixed approximation method by adding flatness constraints for a certain number of derivatives at in the general least squares approximation problem, employing the Lagrange multiplier method for the constraints as in [54] and [107]. An interesting technique for Taylor series approximation of the sinc function was proposed in [109].

Minimax Design of FIR Fractional Delay Filters

Both the least squares and the maximally flat approximation techniques have the drawback that the peak error value in a defined approximation band cannot be controlled explicitly. For example, we may want to design a delay filter whose error characteristics fit in a certain *tolerance scheme*, i.e., it is desired to keep the peak approximation error in given limits. These kinds of specifications can be met using the *minimax (Chebyshev)* solution which, by definition, minimizes the maximum value of the error magnitude in the range of approximation, or

$$E_{\max} = \min\left\{ \max_{\omega\in[0,\alpha\pi]} \left\{ \left| E(e^{j\omega}) \right| \right\} \right\} \tag{44}$$

When certain conditions are met, the minimax solution is unique and equal to the *equiripple* solution, characterized by an oscillating error curve which attains the maxima at a certain number r of frequency points in the approximation interval, i.e.,

$$\left| E(e^{j\omega}) \right| = E_{\max},\ \omega=\omega_k,\ k=1,2,\ldots,r \tag{45}$$

This is the case, for example, in the linear-phase approximation for FIR filters with symmetric or antisymmetric impulse response where the error function reduces to a real-valued cosine series (with a possible weight function). The equiripple solution can be found by the iterative Remez exchange algorithm, as proposed by Parks and McClellan [83]. Unfortunately, Chebyshev approximation problems can usually be solved only by using iterative techniques.

However, the fractional delay approximation is more troublesome than the case of linear-phase FIR filters since the approximating function is complex-valued in general. Advanced algorithms for complex approximation with minimax error characteristics have been presented in [4], [84], [87], [98], and [41] and they can also be applied to the problem at hand. However, as the design procedures involve iterative algorithms, they are not suited for applications requiring real-time coefficient update.

A simplifying formulation for the minimax complex design in terms of real functions was proposed in [88]. The complex task was split into two real-valued design problems which facilitates the use of the efficient Remez algorithm by a modification to the Parks-McClellan program [83]. A new algorithm capable of full complex approximation was proposed in [41]. However, these algorithms are not much better for real-time coefficient update since the Remez routine must be employed for each delay value.

An approach that comes closer to real-time requirements was proposed by Oetken [80]. Considering the design of even-length interpolation FIR filters for sampling rate increase using a polyphase structure, the amplitude errors of the polyphase branches, when made equiripple, are almost exactly proportional to each other. In particular, the zeros of the magnitude error function remain the same with high precision. When the coefficients of one branch are given, those of any other branch can be solved via a set of linear equations. As one of the branches corresponds to an exactly linear-phase filter with a symmetric impulse response, it can be designed off-line using the standard Remez algorithm.

Realizing that each polyphase branch actually approximates a rational fraction of the unit delay, the method can be directly applied to our problem. Let us assume that the coefficients of an Nth-order (or length $L=(N+1)$, L even) symmetric FIR filter are given, the amplitude response of which approximates unity in the equiripple sense in the passband $[0, \alpha\pi]$, $0<\alpha\leq 1$. The error function is known to have $K=L/2$ zeros in the passband, i.e.,

$$E(e^{j\omega}) = H(e^{j\omega}) - H_{id}(e^{j\omega}) = 0,\ \omega=\Omega_k,\ k=1,2,\ldots K \tag{46}$$

implying that

$$\sum_{n=0}^{N} h(n) e^{-jn\Omega_k} = e^{-j\Omega_k N/2}, \; k = 1,2,\ldots,K \tag{47}$$

where $N/2$ is the delay of the filter. Assuming that the zeros remain the same for noninteger values of D as well, the filter coefficients can be solved from Eq. 47 for a chosen total delay D, which is close to $N/2$. This can be expressed in matrix form as

$$\mathbf{E}_\Omega \mathbf{h} = \mathbf{e}_D \tag{48}$$

where

$$\mathbf{E}_\Omega = \begin{bmatrix} 1 & e^{-j\Omega_1} & e^{-j2\Omega_1} & \cdots & e^{-jN\Omega_1} \\ 1 & e^{-j\Omega_2} & e^{-j2\Omega_2} & \cdots & e^{-jN\Omega_2} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & e^{-j\Omega_k} & e^{-j2j\Omega_k} & \cdots & e^{-jN\Omega_k} \end{bmatrix} \tag{49a}$$

is a K x $(N + 1)$ matrix and

$$\mathbf{e}_D = [e^{-jD\Omega_1} \;\; e^{-jD\Omega_2} \;\; \ldots \;\; e^{-jD\Omega_k}]^T \tag{49b}$$

Equation 48 is a set of K complex equations with $L = 2K$ unknowns, which can be expressed as a fully determined set of L real equations by equating the real and imaginary parts of both sides as

$$\mathbf{P}_\Omega \mathbf{h} \equiv \mathbf{p}_\Omega \tag{50a}$$

with

$$\mathbf{P}_\Omega = \begin{bmatrix} \mathbf{C}_\Omega \\ \mathbf{S}_\Omega \end{bmatrix} \tag{50b}$$

and

$$\mathbf{p}_\Omega = \begin{bmatrix} \mathbf{c}_D \\ \mathbf{s}_D \end{bmatrix} \tag{50c}$$

where the matrices and vectors contain appropriate cosine and sine elements such that $\mathbf{E}_\Omega = \mathbf{C}_\Omega - j\mathbf{S}_\Omega$, and $\mathbf{e}_D = \mathbf{c}_D - j\mathbf{s}_D$. Note that the design scheme can be interpreted as a complex version of the *frequency sampling* technique [84] where the frequency samples are unequally spaced.

Hence, one first has to design the linear-phase prototype filter, to find its zero frequencies Ω_k and then to invert the cosine-sine matrix of Eq. 50b. Since the zeroes of the phase error function need not be known with high precision, simple noniterative search on the employed frequency grid is sufficient in general. After that, the coefficients of a new filter approximating any given delay are readily obtained via a single matrix multiplication. Note that matrix Eq. 50b is independent of the delay D and only needs to be inverted once off-line so that the approach is suitable for real-time coefficient update.

Oetken also observed that the amplitude ripple is not the same for all the polyphase branches (in our case: for all fractions of the delay). Instead, it depends on the fractional part d in the following manner [80]:

$$\delta_d = \delta_{1/2} \sin(d\pi) \tag{51}$$

where $\delta_{1/2}$ is the maximum amplitude ripple of the linear-phase prototype filter with the delay $D = N/2$ (= Int(D) + 1/2) and δ_d is the ripple of a filter approximating the (noninteger) delay D with the fractional part, d. Note that the amplitude ripple is largest in the linear-phase case ($D = N/2$ or $d = 0.5$) and reduces to zero when the fraction approaches an integer value ($d = 0$ or $d = 1$), which corresponds to the case that the impulse response reduces to a unit pulse. The almost-equiripple approximation computed using the Oetken method is illustrated in Fig. A6. The 4-tap FIR filter does not have many ripples, but the 10-tap filter responses are seen to be very close to equiripple.

Relation to Interpolation/Decimation FIR Filters

As already discussed above, the *polyphase structure* of decimation/interpolation filters can be utilized for fractional delay implementation with fixed steps ([5, 9, 23, 24, 74]). For example, in order to break the unit delay into Q steps, one can design a Qth-band lowpass filter with the normalized passband width of Q and form the Q-branch polyphase structure by picking up every Qth sample to one branch. It can be shown that each band approximates a fractional delay of the value

$$D = \frac{Q-k}{Q}, \; k = 0,1,2,\ldots,Q-1 \tag{52}$$

The accuracy of approximation depends naturally on the length of the prototype Qth-band lowpass filter. In order to achieve comparable frequency response for each branch, the length of the prototype filter should be a multiple of Q. The lowpass filter should be linear-phase, but it can be designed with any method. However, the optimality of the prototype filter is not shared with the branch filters; e.g., equiripple magnitude characteristic will be lost.

The multirate approach is straightforward and well suited for table look-up applications. If one is satisfied for example, with Q = 50-step division of the unit delay, one simply designs a length-50L filter and uses the desired length-L filter of the set.

However, small delay steps and strict specifications for the approximation error may result in an FIR filter with hundreds of taps, which makes it impossible to use the Parks-McClellan algorithm for the prototype design. In that case, windowing methods may be used for which there is practically no limit for the filter order. However, as there are several other simple methods that provide smaller error, the multirate approach now appears somewhat outdated for this application.

For reference, we chose Q = 10 to design prototype filters of lengths 40 and 100 to provide a set of length-4 and length-10 FIR filters for 10-step increments of the fractional delay. The results are shown in Fig. A7. The startling feature is that the magnitude approximation error is large for the case that the delay is zero. The reason is that the impulse response does not have a single nonzero value as it would have in the ideal case. This can be alleviated by using the "TRICK", which forces exact zeroes in the impulse response [117].

However, rather than for approximation, the multirate approach can be used for implementation of high-quality wideband FD filters, as proposed in [75]. Using a polyphase implementation with two branches the accuracy of approximation can be increased without excessive total delay.

Controlling the Delay of Arbitrary FIR Filters

Above, we have assumed that the fractional delay filter is designed exclusively for producing the desired delay. However, in time-critical applications this may be impossible since an additional FIR filter always introduces some net delay, as discussed earlier. Instead, it may be more advantageous to control the delay of an FIR filter that is already included in the system. As the FIR filter may be adaptive or variable or for other reasons its coefficients may not be known, it is sometimes essential that the delay control algorithm be independent from the filter coefficients.

1) Fourier Transform Based Methods

We first present a Fourier transform based delay control algorithm which is related to ideal bandlimited interpolation discussed in [81]. Let us assume an Nth-order (length $L = N + 1$) prototype filter whose frequency response is

$$H_p(e^{j\omega}) = \sum_{k=0}^{N} h_p(k) e^{-jk\omega} \tag{53}$$

If the filter is linear-phase, it has constant group delay $N/2$. Assume that it is desired to change the delay by a fraction from this nominal value. This is equal to multiplying the frequency response by $e^{-j\Delta d\omega}$ or

$$H(e^{j\omega}) = e^{-j\Delta d\omega} \sum_{k=0}^{N} h_p(k) e^{-jk\omega} = \sum_{k=0}^{N} h_p(k) e^{-j(k+\Delta d)\omega} \tag{54}$$

This is thus the desired frequency response. Taking the symmetric inverse discrete-time Fourier transform (Eq. 11) of Eq. 54, we obtain the real-coefficient FIR filter as

$$\begin{aligned} h(n) &= \frac{1}{2\pi} \int_{-\pi}^{\pi} H(e^{j\omega}) e^{jn\omega} d\omega \\ &= \sum_{k=0}^{N} h_p(k) \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{-j(k+\Delta d)\omega} e^{jn\omega} d\omega \\ &= \sum_{k=0}^{N} h_p(k) \operatorname{sinc}(n-k-\Delta d) \end{aligned} \tag{55}$$

which is seen to be a weighted sum of sinc functions (Eq. 13) with an infinitely long impulse response and consequently with truncation problems. This can be alleviated in the usual manner by employing a suitable time-domain window to truncate $h(n)$. When the lengths of $h(n)$ and $h_p(n)$ are chosen to be the same L, the solution (Eq. 55) can be given in explicit matrix form as

$$\mathbf{h} = \mathbf{W}_{\Delta d} \mathbf{S}_{\Delta d} \mathbf{h}_p \tag{56}$$

where $\mathbf{h}$ and $\mathbf{h}_p$ are the new and the prototype FIR coefficient vectors, respectively, and the elements of the L x L square matrix $\mathbf{S}_\Delta$d are

$$S_{\Delta d,k,l} = \operatorname{sinc}(k-l-\Delta d) \quad k,l = 1,2,\ldots L \tag{57a}$$

and $\mathbf{W}_\Delta$d is a diagonal matrix with the value of the length-L window function as its elements, or

$$W_{\Delta d,k,k} = w_{\Delta d}(k) \quad k = 1,2,\ldots L \tag{57b}$$

Note that the length of the new h vector can also be chosen longer or shorter than L by simply adding rows to or canceling them from the $\mathbf{S}_\Delta$d matrix. Furthermore, if a fixed step Δd is sufficient, one can compute $\mathbf{S}_\Delta$d and multiply it by the $\mathbf{W}_\Delta$d matrix in advance, which considerably reduces the computations.

A similar delay control method employing two discrete Fourier transforms (DFT) was presented by Adams in [2]. It is in fact closely related since, by adapting to our notation, the two complex DFTs can be packed into a single real-coefficient matrix $\mathbf{S}^A_{\Delta d}$ of the form (Eq. 57a) with the elements

$$S^A_{\Delta d,k,l} = \frac{\sin[\pi(k-l-\Delta d)]}{L \sin[\pi(k-l-\Delta d)/L]} \quad k,l = 1,2,\ldots L \tag{58}$$

The Adams method is thus seen to be a discrete-frequency counterpart of the first method, as the essential difference is that the sinc function is replaced by the periodic sampling function, also known as the Dirichlet kernel [11]. The difference in performance of Eq. 57a as compared to Eq. 58 is thus expected to be small for large L. However, both formulas are clearly more advantageous for practical use than the two explicit complex DFTs as proposed in [2] (even if the FFT algorithm is employed), since here only real-valued multiplications are needed. Furthermore, the matrix (Eq. 58)—as well as Eq. 57a—has a symmetric Toeplitz structure, which means that only L different coefficients are required.

For illustration, we approximate the equiripple filters of Fig. A6 using this method. The prototype filter was chosen in the middle ($d = 0.2$) and shifted Dolph-Chebyshev windows with 40 dB ripple were used. From the resulting Fig. A8 it is observed that the magnitude and phase delay responses are poorly preserved for the 4-tap filter but fairly good for the 10-tap filter.

2) Matrix Transform Method Based on Zeros of the Error Function

It is natural to assume that the zeros of the error function remain the same not only independently of the approximation method but also irrespective of the amplitude curve approximated. This suggests that the Oetken method can be used to control the delay of any FIR filter. This is, however, only possible when the zeros of the error function are known beforehand so that the transformation matrix (which only depends on the zeros) can be computed.

3) Farrow Structure for Fractional Delay FIR Filters

A promising technique for efficient implementation of a continuously variable delay element was proposed by Farrow [29]. This method assumes that the filter is designed off-line, but the real-time control of the delay value is simple and efficient. The basic idea is to design a set of filters approximating a fractional delay in the desired range (e.g., $0 \leq d \leq 1$) and then to approximate each coefficient as a Pth-order polynomial of d, or

$$h_d(n) = \sum_{m=0}^{P} c_m(n) d^m \quad n = 0, 1, 2, \ldots N \tag{59}$$

where $c_m(n)$ are real-valued approximating coefficients. The subscript d is now included to emphasize that each coefficient is a function of the fractional delay, d. The transfer function of the filter can be elaborated into the form

$$\begin{aligned} H_d(z) &= \sum_{n=0}^{N} h_d(n) z^{-n} = \sum_{n=0}^{N} \left[\sum_{m=0}^{P} c_m(n) d^m \right] z^{-n} \\ &= \sum_{m=0}^{P} \left[\sum_{n=0}^{N} c_m(n) z^{-n} \right] d^m = \sum_{m=0}^{P} C_m(z) d^m \end{aligned} \tag{60}$$

where it was defined

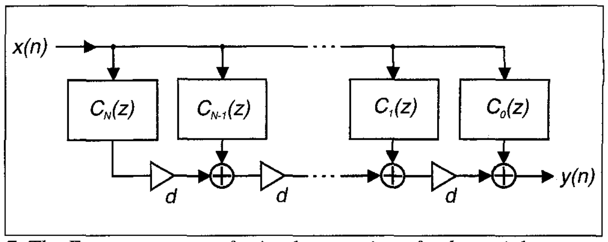
$$C_m(z) = \sum_{n=0}^{N} c_m(n) z^{-n} \tag{61}$$

The form (Eq. 60) immediately suggests an efficient implementation as a parallel connection of fixed filters with output taps weighted by an appropriate power of d (Fig. 7).

The sample implementation presented in [29] was designed by employing least squared error criterion over the desired frequency band and the employed range of d. Also the polynomial approximation of the filter coefficients was done in the least squares sense. An excellent tutorial presentation of the method with examples and performance analysis is included in [72], when applied to timing adjustment algorithms in digital receivers.

The polynomial approach can readily be generalized for other filter design techniques as well. In addition to the least squares method, maximally flat or equiripple approximation can be employed to design the set of prototype filters covering the desired range of d. The coefficients of the filter set are then approximated separately by the polynomial structure of a desired order. Polynomial interpolation techniques, such as Lagrange interpolation, can be realized using the Farrow structure without further approximation [27], [126], [130], [135].

The design of the generalized polynomial structure for delay control can be formulated as follows:



7. The Farrow structure for implementation of polynomial approximation of filter coefficients.

1) Design a set of $Q + 1$ FIR filters of the (same) chosen order N approximating in a desired sense the ideal noninteger delay whose fractional part takes values in the desired range [$d_{\min}$, $d_{\max}$]. The values of d can be chosen, for example, on a uniform grid as

$$d_q = q \frac{d_{\max} - d_{\min}}{Q}, \quad q = 0, 1, \ldots Q \tag{62}$$

The result is the set of prototype filters with the coefficients $h_{d,q}(n)$ for $q = 0, 1, 2, \ldots, Q$ and $n = 0, 1, 2, \ldots, N$.

2) Design the polynomial structure approximating the coefficients of the prototype filters in the desired sense so that

$$h_{d,q}(n) = \sum_{m=0}^{P} c_m(n) d_q^m, \quad q = 0, 1, 2, \ldots, Q; \; n = 0, 1, 2, \ldots, N \tag{63}$$

Note that the design reduces to $L = N + 1$ separate optimization problems: each polynomial approximation is carried out for a fixed value of n. It is easiest to use least squares curve fitting for this task. According to our experience, second-order polynomials with three coefficients are usually sufficient. The resulting coefficients $c_m(n)$ are then employed to form the transfer functions $C_m(z)$ of the subsections, as shown in Fig. 7.

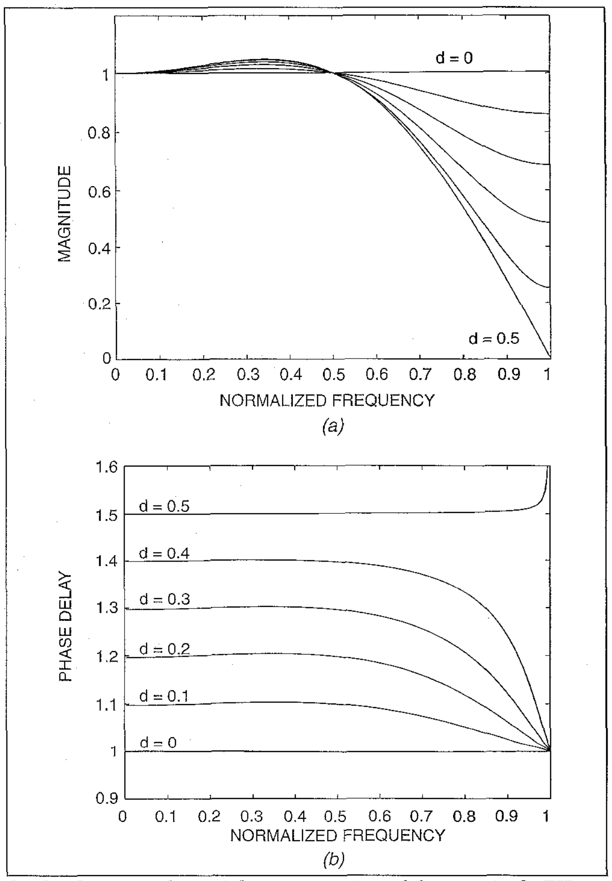
We applied the design method to imitate the response of 4-tap equiripple approximations of Fig. A6. Second-order polynomials ($P = 2$) were used for coefficient approximation. The results are shown in Figs. 8a and 8b, which show that both the amplitude and phase delay characteristics are accurately reproduced, with hardly any observable deviation from the original ones (Figs. A6a and b). Similar results were obtained for 10-tap filters. Farrow approximation with second-order polynomial approximation for coefficients thus provides an accurate means for practical implementation of FIR FD filters.

The polynomial approximation technique can also be applied to allpass filters, as will be discussed shortly.

Summary of FIR Filter Design and Implementation

To conclude our discussion of FIR FD filters, we present a summary of these techniques and evaluate their design complexity. As stated previously, our main interest is in fast on-line tuning of the fractional delay and thus the fast update of the coefficients is of paramount importance.

Table: Design Methods for FIR Fractional Delay Filters			
Method	**Parameters**	**Design**	**Comments**
A. LS-Based FIR Design			
0) Ideal Sinc	N D	Closed-form	For theoretical use only
1) LS FIR (Truncated Sinc)	N D	Closed-form	Gibbs phenomenon
2) Windowed Sinc	N D window	Closed-form	
3) Smooth Transition Functions	N D transition func	Closed-form	
4) Complex LS Design	N D	Matrix Eq.	
5) Discrete Complex LS Design	N D grid	Matrix Eq.	
6) Stochastic LS Design	N D signal spectrum	Matrix Eq.	Utilizes the power spectrum of the signal
B. Maximally Flat Design			
Lagrange Interpolation	N D	Closed-form	Simple design: try this first
C. Equiripple Design			
Oetken Method	N D prototype FIR filter	Matrix Eq.	
D. Multirate Method			
Polyphase Design	N D FIR Filter	High-order FIR design	
E. FIR Delay Control			
1) Fourier-Transform Method	N D prototype FIR filter	Matrix Mult.	
2) General Oetken Method	N D FIR filter + zeros	Matrix Eq.	
3) Farrow Structure	N D P set of FIR filters	Polynomial approximation	easy and accurate delay control



8. The Farrow polynomial approximation of the equiripple FIR design of A6. Polynomial order $P = 2$ and filter length $L = 4$. a) magnitude and b) phase delay response.

The above discussed FIR design methods are collected in the Table above, with information about the filter parameters and design complexity. The Lagrange interpolation is one of the most attractive methods, since only a small number of multiplications and additions is needed for coefficient update [52]. Filters of the order 1, 2, and 3 are fast to compute and accurate enough for many applications [42, 120, 121, 125]. Often, the coefficients must be updated only for every 5th to 50th signal sample so that relatively many operations may be spent for each update.

The Farrow polynomial approximation of filter coefficients [29] also offers means for fast coefficient update. Since any approximation technique can be used for the prototype design, polynomial approximation is a highly flexible tool and suitable also for nonstandard applications, where, for example, an irregular magnitude response is to be maintained.

In more complicated filter formulations and design algorithms, real-time coefficient update may be too expensive unless table lookup techniques are utilized [103, 106]. This strategy is very efficient when the table is precompiled so that proper filter coefficients can be retrieved by table lookup for a finite set of fractional delay values or additional interpolation between stored table values.

Fractional Delay Approximation Using Allpass Filters

In general, a recursive (IIR) digital filter can meet the same frequency-domain specifications with a smaller number of multiplications than an FIR filter. Unfortunately, the design of IIR filters with prescribed magnitude and phase (or group delay, or phase delay) response is far more complicated than that of corresponding FIR filters. The design of FIR filters is greatly eased by the fact that the filter coefficients are equal to the samples of the filter impulse response so that (in full-band approximation) the frequency-domain specifications can be turned into the "coefficient domain" by the inverse discrete-time Fourier transform. This is not possible for recursive filters.

Another major disadvantage is the possible *instability* of recursive filters. In general, the obtained solution has to be checked so that all the poles of the filter remain within the unit circle in the z-domain, which makes real-time coefficient update difficult.

Here we omit the problem of magnitude approximation by considering only the design of allpass filters, a special subclass of recursive filters. (For a report on fractional delay IIR filter design, see [113].) Allpass filters have unity magnitude response in the whole frequency band by definition, which means that one can concentrate on the approximation of the desired phase (or group delay, or phase delay) characteristics. This reduces the available degrees of freedom but also makes the design task much easier.

The z transfer function of an Nth-order allpass filter is of the form

$$A(z) = \frac{z^{-N} D(z^{-1})}{D(z)} = \frac{a_n + a_{n-1} z^{-1} + \ldots + a_1 z^{-(N-1)} + z^{-N}}{1 + a_1 z^{-1} + \ldots + a_{N-1} z^{-(N-1)} + a_n z^{-N}} \qquad (64)$$

where the numerator polynomial is a mirrored version of the (supposedly stable) denominator $D(z)$. The coefficients are assumed to be real-valued. The direct form I implementation of the allpass filter is shown in Fig. 9. The *phase response* of the allpass filter can be expressed as

$$\Theta_A(\omega) = \arg\left\{A(e^{j\omega})\right\} = -N\omega + 2\Theta_D(\omega) \qquad (65)$$

where

$$\Theta_D(\omega) = \arg\left\{\frac{1}{D(e^{j\omega})}\right\} = \arctan\left\{\frac{\sum_{k=0}^{N} a_k \sin(k\omega)}{\sum_{k=0}^{N} a_k \cos(k\omega)}\right\} = \arctan\left\{\frac{\mathbf{a}^T\mathbf{s}}{\mathbf{a}^T\mathbf{c}}\right\} \qquad (66)$$

where $\mathbf{c}$ and $\mathbf{s}$ are appropriate cosine and sine vectors as defined in Eq. 30, and $\mathbf{a}$ is the coefficient vector

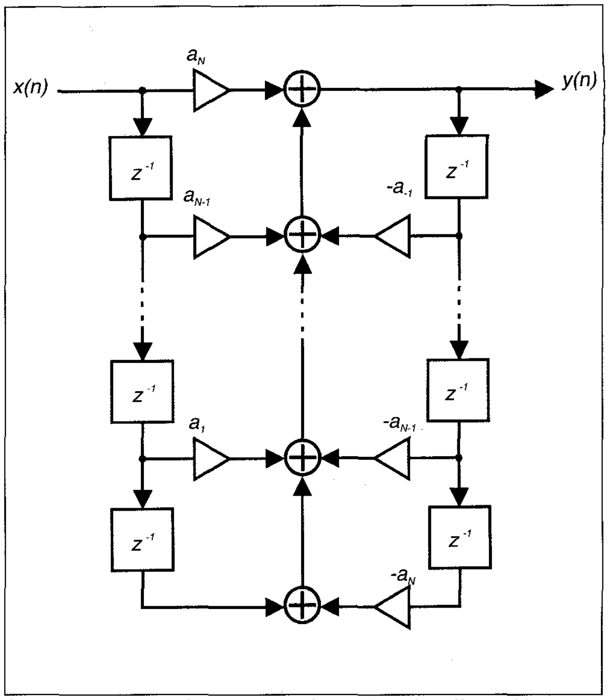
$$\mathbf{a} = [a_0 \ a_1 \ a_2 \ \ldots \ a_N]^T \qquad (67)$$

with $a_0 = 1$. The *group delay* of the allpass filter is related to that of the denominator similarly to Eq. 65, or

$$\tau_{g,A}(\omega) = -\frac{d\Theta_A(\omega)}{d\omega} = N - 2\tau_{g,D}(\omega) \qquad (68)$$

where the delay of the denominator can be expressed as

$$\tau_{g,D}(\omega) = -\frac{d\Theta_D(\omega)}{d\omega} = \frac{\mathbf{a}^T\mathbf{G}\Lambda\mathbf{a}}{\mathbf{a}^T\mathbf{G}\mathbf{a}} \qquad (69a)$$



9. Direct form I implementation of an Nth-order allpass filter.

where

$$\mathbf{G} = \mathbf{c}\mathbf{c}^T + \mathbf{s}\mathbf{s}^T \qquad (69b)$$

$$\Lambda = \mathrm{diag}[0 \ 1 \ \ldots \ N] \qquad (69c)$$

Naturally, for the *phase delay* it holds

$$\tau_{p,A}(\omega) = -\frac{\Theta_A(\omega)}{\omega} = N - 2\tau_{p,D}(\omega) \qquad (70)$$

Unfortunately, the phase, phase delay, and group delay are all related to the filter coefficients in a very nonlinear manner, as the above equations show. This means that one cannot expect as simple design formulas for the allpass filter coefficients as for FIR filters. Instead, one can almost exclusively find only iterative optimization techniques for minimization of traditional error criteria.

In the following discussion, we shall review in greater detail only the simplest allpass design techniques that have some potential for applications requiring real-time coefficient update. Among the above delay measures, the phase is perhaps the most suitable for least squared error design. Consequently, much of the following is based on the recent results on least squares phase approximation presented in [60, 63, 53, 77]. These schemes are easy to program and can also be modified for approximately equiripple phase error solutions, as well as for corresponding phase delay approximations, in contrast to many other methods that may be more difficult to use.

Least Squares Design of Allpass Filters

1) Approximate LS Phase Error Design

Using the above notation, the phase error (deviation from a prescribed desired phase $\Theta_{id}(\omega)$) can be expressed as

$$\Delta\Theta(\omega) = \Theta_{id}(\omega) - \Theta_A(\omega) = 2\arctan\left\{\frac{\mathbf{a}^T\mathbf{s}_\beta}{\mathbf{a}^T\mathbf{c}_\beta}\right\} \tag{71}$$

where

$$\mathbf{s}_\beta = [\sin\{\beta(\omega)\}\ \sin\{\beta(\omega)-\omega)\}\ \dots\ \sin\{\beta(\omega)-N\omega\}]^T$$
$$\mathbf{c}_\beta = [\cos\{\beta(\omega)\}\ \cos\{\beta(\omega)-\omega)\}\ \dots\ \cos\{\beta(\omega)-N\omega\}]^T \tag{72,73}$$

and

$$\beta(\omega) = \frac{1}{2}[\Theta_{id}(\omega) + N\omega] \tag{74}$$

When approximating a noninteger delay $D = N + d$, or $\Theta_{id}(\omega) = -D\omega = -(N+d)\omega$, the last expression reduces to

$$\beta(\omega) = -\frac{\omega d}{2} \tag{75}$$

The term $N\omega$ is canceled out, since the average delay of the system is exactly N samples. In [60, 63, 53, 77], several techniques were developed to minimize the weighted least squared phase error, i.e., the measure

$$E = \frac{1}{\pi}\int_0^{\alpha\pi} W(\omega)|\Delta\Theta(\omega)|^2\, d\omega \tag{76}$$

where $W(\omega)$ is a nonnegative weight function. By using linear approximation for the arcus tangent function $\arctan(x) \approx x$ in the expression for the phase error (Eq. 71), a modified error measure can be expressed as

$$E = \frac{1}{\pi}\int_0^{\alpha\pi} W(\omega)\frac{|2\mathbf{a}^T\mathbf{s}_\beta|^2}{|\mathbf{a}^T\mathbf{c}_\beta|^2}d\omega = \frac{4}{\pi}\int_0^{\alpha\pi} W(\omega)\frac{\mathbf{a}^T\mathbf{s}_\beta\mathbf{s}_\beta^T\mathbf{a}}{\mathbf{a}^T\mathbf{c}_\beta\mathbf{c}_\beta^T\mathbf{a}}d\omega$$
$$= \frac{4}{\pi}\int_0^{\alpha\pi} W(\omega)\frac{\mathbf{a}^T\mathbf{S}_\beta(\omega)\mathbf{a}}{\mathbf{a}^T\mathbf{C}(\omega)\mathbf{a}}d\omega \tag{77}$$

where we define new matrices as $\mathbf{S}_\beta(\omega) = \mathbf{s}_\beta\mathbf{s}_\beta^T$ and $\mathbf{C}(\omega) = \mathbf{c}_\beta\mathbf{c}_\beta^T$. If the coefficient vector a in the denominator were known and fixed, this error measure would be a quadratic form expressible as

$$E = \mathbf{a}^T\left[\frac{4}{\pi}\int_0^{\alpha\pi} W(\omega)\frac{\mathbf{S}_\beta(\omega)}{\mathbf{a}_0^T\mathbf{C}(\omega)\mathbf{a}_0}d\omega\right]\mathbf{a} = \mathbf{a}^T\mathbf{P}\mathbf{a} \tag{78}$$

where $\mathbf{a}_0$ is the fixed coefficient vector and the matrix $\mathbf{P}$ is defined in an obvious manner. When the matrix $\mathbf{P}$ is positive definite, there exists a unique solution for the vector $\mathbf{a}$ which minimizes the error measure (Eq. 78). The solution can be found, e.g., by the eigenfilter technique [116] which is equivalent to iteratively solving a set of N linear equations (Appendix C).

The easiest way to eliminate the denominator is to neglect it, which effectively introduces coefficient-dependent weighting in the error measure and thus causes a bias from the true least squared error solution. In this case the matrix $\mathbf{P}$ is solved as

$$\mathbf{P} = \frac{4}{\pi}\int_0^{\alpha\pi} W(\omega)\mathbf{S}_\beta(\omega)d\omega \tag{79}$$

Due to the simple form of $\beta(\omega)$, the integrals can be solved in closed form, e.g., if $W(\omega)$ is chosen piecewise constant. When it is set to $W(\omega) = 1$ in the approximation band $\omega \in [0, \alpha\pi]$, the elements of the $\mathbf{P}$ matrix are obtained as

$$P_{k,l} = \frac{4}{\pi}\int_0^{\alpha\pi}\{\cos[(k-l)\omega] - \cos[(N-(k+l+d)\omega]\}d\omega$$
$$= 4\alpha\{\mathrm{sinc}[\alpha(k-l)] - \mathrm{sinc}[\alpha(N-(k+l+d))]\}$$
$$k,l = 1,2,\dots,L \tag{80}$$

The matrix is seen to have a Toeplitz-plus-Hankel structure. This approximation scheme offers an efficient way to solve for the allpass coefficients, only requiring solution of the set of N equations. Using fast algorithms like the one proposed in [71], this matrix can be inverted in order of N^2 arithmetic operations instead of N^3 complexity required for inversion of a general matrix.

The bias from the least squares solution can be removed by employing an iterative algorithm such that the old coefficients are used in the denominator for weighting, as originally proposed in [60] and [63]. With this scheme, the matrix at the qth iteration is

$$\mathbf{P}^{(q)} = \frac{4}{\pi}\int_0^{\alpha\pi} W(\omega)\frac{\mathbf{S}_\beta(\omega)}{\mathbf{a}^{(q-1)T}\mathbf{C}_\beta(\omega)\mathbf{a}^{(q-1)}}d\omega \tag{81}$$

For details, see [63] or [77]. As demonstrated in these references, the algorithm typically converges to the desired solution although it cannot be guaranteed.

Let us demonstrate both methods by approximating fractional delay with second and fifth-order allpass filters. The phase delay curves obtained using the noniterative method (Eq. 79) are shown in Figs. B1a and b, respectively. Note that we show the phase delay responses for the entire range $-0.5 \leq d \leq 0.5$ since, unlike FIR filters, no symmetry relations hold for allpass filters. The phase delay approximation appears to be worse at low frequencies. This is because the filter is designed by minimizing the phase error but the phase delay (phase divided by frequency) naturally yields large values when the frequency is small. Furthermore, it was observed that the iterative method (Eq. 81) produces essentially equivalent results so that, in contrast to phase equalizers [77], in FD filter design the noniterative method should be preferred in practice.

2) LS Phase Delay Error Design of Allpass Filters

Since the phase delay is defined as the negative phase divided by angular frequency (Eq. 10), the LS phase delay error can be expressed in terms of the phase error as follows

$$E=\frac{1}{\pi}\int_0^{\alpha\pi} W(\omega)|\Delta\tau_p(\omega)|^2\, d\omega=\frac{1}{\pi}\int_0^{\alpha\pi} W(\omega)\left|\frac{\Delta\Theta(\omega)}{\omega}\right|^2 d\omega$$
$$=\frac{1}{\pi}\int_0^{\alpha\pi} \frac{W(\omega)}{\omega^2}|\Delta\Theta(\omega)|^2\, d\omega \qquad (82)$$

Hence, the above phase error solutions can be modified for LS phase delay design by simply introducing an additional weighting function $W'(\omega) = 1/\omega^2$. For example, in this case the **P** matrix of Eq. 80 becomes

$$P_{k,l}=\frac{4}{\pi}\int_0^{\alpha\pi}\frac{1}{\omega^2}\{\cos[(k-l)\omega]-\cos[(N-(k+l+d))\omega]\}d\omega$$
$$k,l=1,2,\dots,L \qquad (83)$$

whose elements can be determined either numerically or by employing the relation

$$\int\frac{\cos(ax)}{x^2}=-a\,\mathrm{Si}(ax)-\frac{\cos(ax)}{x} \qquad (84)$$

where the sine integral $\mathrm{Si}(x)$ has a fast converging series expansion [1]:

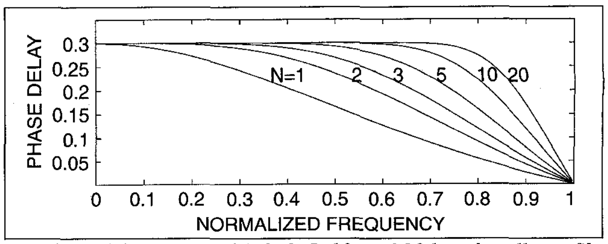
$$\mathrm{Si}(x)=\int_0^x\frac{\sin t}{t}dt=\sum_{n=0}^{\infty}\frac{(-1)^n x^{2n+1}}{(2n+1)(2n+1)!} \quad t\ge 0 \qquad (85)$$

The iterative approach is treated similarly. However, as with phase approximation, also here the iterative solution produced practically the same results, and thus there is no need for the additional complication in FD filter design. The phase delay curves obtained using the noniterative phase delay approximation technique are shown in Fig. B2. Comparison to phase approximation (Fig. B1) demonstrates that the phase delay approximation (Fig. B2) is more appropriate when the design is judged by the phase delay curves.

Maximally Flat Group Delay Design of Allpass Filters

In 1971, Thiran proposed an analytic solution for an all-pole lowpass filter with a maximally flat group delay response at the zero frequency [114]. Since the group delay of an allpass filter is twice that of the corresponding all-pole filter, the all-pole formulas of Thiran can be used for allpass design by using half of the delay value. (In fact, the Thiran formulas appear to be much more useful for allpass design, since there is no way to control the amplitude response of the all-pole lowpass filter.) The solution for the allpass filter coefficients approximating the delay $D = N + d$ is

$$a_k=(-1)^k\binom{N}{k}\prod_{n=0}^{N}\frac{D-N+n}{D-N+k+n} \quad \text{for } k=0,1,2\dots,N \qquad (86)$$



10. Phase delay curves of 1, 2, 3, 5, 10, and 20th-order allpass filters approximating fractional delay d = 0.3 (maximally flat group delay approximation; only fractional part of the delay shown).

where $\binom{N}{k}=\frac{N!}{k!(N-k)!}$ is a binomial coefficient. It always holds that $a_0 = 1$ so that the polynomial is automatically scaled as desired.

Figure 10 shows the phase delay curves of 1, 2, 3, 5, 10, and 20th-order allpass filters approximating the fractional delay value $d = 0.3$. Note that the integer parts of the delays are different. The corresponding phase delay responses for filter orders $N = 2$ and $N = 5$ are presented in Fig. B3. It is seen that, even with such low-order filters, the delay response is excellent over a large part of the frequency band.

In the original paper [114] it was shown that for large enough positive D the resulting allpass filter is guaranteed to be stable, which is a great advantage. As this is the only solution known to us where the coefficients of the allpass filter are obtained in closed form, it seems to be the best choice for many practical applications.

Minimax or Equiripple Design of Allpass Filters

There exist numerous algorithms that can be used for minimax or equiripple phase, group delay, or phase delay approximation, e.g., [26, 28, 94, 40, 100, 38, 59, 53]. All these algorithms are iterative and some of them require a good initial solution to converge. Instead of reviewing all the algorithms in detail, we introduce a straightforward approach for equiripple phase approximation which can be implemented and used without too much expertise in the approximation theory. For a comprehensive monograph on phase approximation with allpass filters, see [62].

1) Equiripple Phase Error Design

In [65], an approach for iterative weighting was proposed for approximately equiripple amplitude design of linear-phase FIR filters. In [53] and [77], this technique was applied to least squares phase approximation with allpass filters. The basic idea is to use a weighting function in Eq. 81 that always reduces the maxima of the error curve. This kind of weighting can be constructed by taking an envelope (i.e., a curve connecting local maxima) of the phase error function itself. By raising this envelope curve to a pth power, the following weighting function results:

$$W(\omega)=[\mathrm{env}\,|\Delta\Theta^{(q-1)}(\omega)|]^p \qquad (87)$$

where env() is the envelope function. As seen from Eqs. 76

Design Guide 2: Simplest Allpass FD Filter—Thiran Approximation

The Thiran method can be viewed as a recursive counterpart of Lagrange interpolation. It is the simplest way to design an allpass filter approximating a given fractional delay *D*. It is characterized by maximally flat group delay at the zero frequency [80]. The coefficients are obtained in closed form from Eq. (86), or

$$a_k = (-1)\binom{N}{k}\prod_{n=0}^{N}\frac{D-N+n}{D-N+k+n} \text{ for } k = 0,1,2,\dots,N$$

where *N* is the order of the allpass filter. The denominator coefficients of low-order allpass transfer functions (N = 1, 2, and 3) are given in the Table.

The plots of the phase delay responses of Thiran allpass filters of order N = 1, 2, 3, 5, 10, and 20 in the worst-case approximation (half-sample delay) are presented in Fig. 10—remember that the magnitude response of an allpass filter is always exactly unity in the whole frequency band, by definition. It is seen that these low-order filters give an excellent phase delay approximation at low frequencies. However, the approximation bandwidth grows very slowly when the filter order is increased. If a low-order Thiran allpass filter (e.g., the third-order one) is not good enough for the application, it may be better to use an LS-based allpass filter design method instead.

Table : Coefficients of the Thiran FD Allpass Filters of Order N = 1, 2, and 3.

	a_1	a_2	a_3
$N = 1$	$(1 - D)(1 + D)$		
$N = 2$	$-2(D - 2)/(D + 1)$	$(D - 1)(D - 2)/(D + 1)(D + 2)$	
$N = 3$	$-3(D - 3)/(D + 1)$	$3(D - 2)(D - 3)/(D + 1)(D + 2)$	$-(D - 1)(D - 2)(D - 3)/(D + 1)(D + 2)(D + 3)$

and 87, this solution yields an approximately L_{p+2}-norm solution, since the weighting function effectively increments the power of $\Delta\Theta(\omega)$ by p from the original L_2-solution. When a high enough p is chosen (e.g., $p = 50$), the error curve is very close to equiripple. As discussed in [53] and [77], it is advantageous to increase p in steps larger than one to guarantee fast convergence.

2) Equiripple Phase Delay Error Design

As with least squared error design, the equiripple phase error algorithm can be modified for phase delay design by using the envelope of the phase delay error for weighting in Eq. 82. This results in the following overall weight function at the qth iteration:

$$W(\omega) = \frac{1}{\omega^2}\left[\operatorname{env}\left|\frac{\Delta\Theta^{(q-1)}(\omega)}{\omega}\right|\right]^p \tag{88}$$

Figure B4 provides examples of the equiripple phase delay design using the iterative algorithm, showing that the resulting error behavior is very close to equiripple. The design algorithm required ca. 15-30 iteration steps, depending on the fractional delay value d.

Controlling the Delay of Allpass Filters

Unlike FIR filters, where the control of the delay of an unknown filter is possible, the corresponding devices for allpass filters assume knowledge about a prototype allpass filter. The methods are in general more complicated than those for FIR filters, but nevertheless simpler than the iterative design methods.

1) Matrix Transform Method Based on Zeros of the Error Function

The phase delay curves of Fig. B4b indicate that the zeros of the phase error function remain approximately the same when the fractional delay is changed from -0.5 to 0.5. This suggests an update algorithm similar to that proposed by Oetken [80] where the new coefficients are obtained from the prototype filter via a transformation matrix. Assume that the zeros of the phase error curve in the approximation band $\omega \in [0, \alpha\pi]$ (also bandpass approximation is possible) are Ω_k, $k = 1, 2, \dots, N$. Setting the phase response Eq. 65 of the filter to approximate desired values at these points results in an interpolation which has a unique solution for an Nth-order allpass filter (for details, see [63]). Interpolating the phase response $-D\omega = -(N + D)\omega$ yields the set of equations

$$\sum_{l=1}^{N} a_l \sin[(d+l)\Omega_k] = -\sin(\Omega_k d) \quad k = 1,2,\dots,N \tag{89}$$

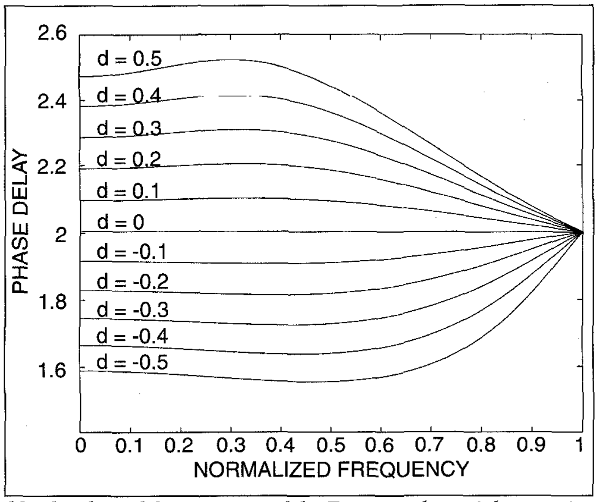
which can be given in matrix form as

$$\mathbf{S}_d \mathbf{a}_1 = -\mathbf{s}_d \tag{90}$$

where

$$S_{d,k,l} = \sin[(d+l)\Omega_k] \quad k,l = 1,2,\dots,N \tag{91a}$$

$$s_{d,k,l} = \sin(\Omega_k d) \quad k = 1,2,\dots,N \tag{91b}$$



11. The phase delay response of the Farrow polynomial approximation of the LS phase delay design of Fig. B2. Polynomial order $P = 2$, filter order $N = 2$, and $\omega_p = 0.5\pi$.

and $\mathbf{a}_1$ equals $\mathbf{a}$ of Eq. 67, where the first element, a_0, has been canceled. Hence, the coefficients of the allpass filter approximating any delay value d can be solved from Eq. 90, via matrix inversion when the zeros of the phase error are known. Unlike the Oetken method, the coefficients of the prototype filter are not required explicitly, since all the necessary information is included in the zeros of the error curve and in the delay value (provided, of course, that the filter order remains the same).

Unfortunately, the matrix $\mathbf{S}_d$ depends on d, so that the inverse matrix cannot be computed beforehand in general as with Oetken's method. Instead, the matrix has to be inverted for each delay value, which is fortunately easier than applying an iterative algorithm every time. In principle, any approximation technique and error measure— phase, group delay, or phase delay error—can be used to determine the prototype filter. For example, it is also possible to design an allpass phase equalizer for an IIR filter and fine-tune its net delay by the proposed method.

2) Recursive Farrow Structure for Allpass Filters

Analogous to the FIR case discussed previously, each coefficient of an allpass filter approximating a fractional delay in the range varies continuously and can be approximated by a polynomial in d. Figure 11 shows the phase delay curves of the polynomial coefficient approximations imitating the least squares phase delay designs of Fig. B3a. Second-order polynomials were used which resulted in practically identical designs. This observation holds for higher-order filters as well. The filter can be implemented efficiently using a recursive version of the Farrow structure that was described previously for FIR filters.

Summary of Allpass Filter Design and Implementation

Although there are far less different approaches for allpass than FIR filter design, the basic conclusions are much the same. A summary of the allpass filter design techniques is collected in the Table below. The maximally flat group delay (Thiran) design corresponds to the Lagrange interpolation: it is a closed-form design with an excellent accuracy of approximation at low frequencies and thus it is the best choice to begin with. A potential drawback is the division required in the computation of the coefficients, which may be slow and difficult to implement, e.g., with digital signal processors. If more sophisticated optimization is desired, other approximation techniques can be used and the filter can be implemented by employing the Farrow polynomial structure or table lookup methods.

Choosing the Right Method

There is a plethora of methods available for engineers willing to apply fractional delay filters, and one may be confused about which one to pick. This is a generic problem in many engineering tasks. We suggest the following general principle: try simple methods first. This means that, particularly if one is not quite sure about how to state the magnitude and phase delay constraints (which is often the case), one should try Lagrange interpolation with a few taps, e.g., $L = 2$ or 4—remember that the phase delay response is better for even-length filters. For a comparison of FIR FD filters, see also Cain, *et al.* [15]. If allpass filters are preferred, first or second-order maximally flat group delay (Thiran) filters are recommended. According to our experience, Lagrange FIR and Thiran allpass filters are safe solutions in most cases.

Table: Design Methods for Allpass Fractional Delay Filters

Method	Parameters	Complexity	Comments
A. LS-Based Allpass Design			
1) LS Phase Design	*N D*	Matrix Eq.	Noniterative design
2) LS Phase Delay Design	*N D*	Matrix Eq.	Noniterative design
B. Maximally Flat Design			
Thiran Method		Closed-form	Simple design: try this first
C. Equiripple Design			
Weighted LS Design	*N D*	Iterative	
D. Allpass Delay Control			
1) Modified Oetken Method	*N D* prototype filter + zeros	Matrix Eq.	
2) Recursive Farrow Method	*N D P* and set of allpass filters	Polynomial approx.	Easy delay control

On the other hand, one may want to try to find the best solution for the particular application (e.g., in terms of low implementation complexity, good magnitude and phase delay response in the specified frequency band, or flexible on-line tuning of the delay). To ease the choice we have collected the essential FIR and allpass FD filter design methods in Tables 1 and 2.

How do we make the basic choice between FIR and allpass filters? As in the general approximation problem, a recursive filter meets the magnitude and phase (delay) specifications with a smaller number of multiplications. However, there may be additional finite wordlength problems due to *roundoff noise*, *limit cycles*, and possible *instability* in coefficient quantization (particularly when the coefficients are changed on-line). Furthermore, one may encounter transient problems in real-time tuning applications [105]. The *elimination of transients* in variable-coefficient fractional delay allpass filters has been addressed in [132-135].

Conclusions

In this article, we have addressed the general problem of approximation of a delay that is a fractional part of the sampling interval. Both FIR and allpass filter design techniques have been reviewed. We have evaluated various optimization criteria and design techniques from the practicing engineer's point of view and have tried to provide a good tutorial in the topic. The fractional delay approximation is a generic problem which is encountered in several fields and applications of DSP.

A set of MATLAB programs for FD filter design is available via WWW from http://www.hut.fi/HUT/Acoustics/fdtools.html or via FTP from helmholtz.hut.fi (130.233.160.51) using anonymous login (directory pub/fdtools).

Acknowledgments

This work has been in preparation for many years and several people have contributed to its success. Prof. Gerry Cain brought valuable references to our attention. Dr. Floyd Gardner as well as Prof. Heinrich Meyr and Dr. Gerd Ascheid generously provided advance manuscripts of his paper [32] and of their book [72], respectively. The following individuals provided insightful and helpful comments, which is gratefully acknowledged here: Dr. Tsuhan Chen, Mr. Pertti Järvensivu, Mr. Kari Järvinen, Dr. Kari Kalliojärvi, Dr. Peter Kootsookos, Dr. Markus Lang, Prof. Henrique Malvar, Mr. Dana Massie, Prof. Truong Nguyen, and Prof. Seppo Ovaska. Thanks to Mrs. Elspeth Murphy for checking the language of the manuscript. Also the anonymous reviewers provided many insightful comments which improved the quality of the article. Special thanks are due to Dr. Jonathan Mackenzie for fruitful discussions on the title of this article.

T. I. Laakso is with the University of Westminster, School of Electronic and Manufacturing Systems Engineering, London, U.K., and Helsinki University of Technology, Laboratory of Signal Processing and Computer Technology, Espoo, Finland, on leave from Nokia Research Center, Helsinki, Finland. V. Välimäki, M. Karjalainen, and U.K. Laine are with Helsinki University of Technology, Laboratory of Acoustics and Audio Signal Processing, Espoo, Finland.

References

1. M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions*. New York: Dover Publications, 1972.

2. J. W. Adams, Alternate approach to digital phase shift filters, in *Proc. Int. Symp. Signal Processing: Theories, Implementations and Applications* (ISSPA-87), pp. 160-165, Brisbane, Australia, Aug. 24-28, 1987.

3. 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.

4. A. Alkhairy, K. Christian, and J. Lim, Design of FIR filters by complex Chebyshev approximation, in *Proc. IEEE Int. Conf. Acoust. Speech Signal Processing* (ICASSP-91), vol. 3, pp. 1985-1988, Toronto, Canada, May 14-17, 1991.

5. R. Ansari and B. Liu, Efficient sampling rate conversion using recursive (IIR) digital filters, *IEEE Trans. Acoust. Speech Signal Processing*, vol. 31, no. 6, pp. 1366-1373, Dec. 1983.

6. R. Ansari and B. Liu, Multirate signal processing, in S. K. Mitra and J. F. Kaiser (eds.), *Handbook of Digital Signal Processing*. New York: John Wiley & Sons, 1993, pp. 981-1084.

7. J. Armstrong and D. Strickland, Symbol synchronization using signal samples and interpolation, *IEEE Trans. Communications*, vol. 41, no. 2, pp. 318-321, Feb. 1993.

8. G. Ascheid, M. Oerder, J. Stahl, and H. Meyr, An all-digital receiver architecture for bandwidth efficient transmission at high data rates, *IEEE Trans. Communications*, vol. 37, pp. 804-813, Aug. 1989.

9. 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 Processing*, vol. 24, no 2, pp. 199-126, April 1976.

10. K. Bucket and M. Moeneclaey, The effect of interpolation on the BER performance of narrowband BPSK and (O)QPSK on Rician-fading channels, *IEEE Trans. Communications*, vol. 42, no. 11, pp. 2929-2953, Nov. 1994.

11. C. S. Burrus and T. W. Parks, *DFT/FFT and Convolution Algorithms*. New York: John Wiley & Sons, 1985.

12. C. S. Burrus, A. W. Sowieto, and R. A. Gopinath, Least squared error FIR filter design with transition bands, *IEEE Trans. Signal Processing*, vol. 40, no. 6, pp. 1327-1340, June 1992.

13. C. S. Burrus, Multiband least squares FIR filter design, *IEEE Trans. Signal Processing*, vol. 43, no. 2, pp. 412-421, Feb. 1995.

14. G. D. Cain and A. Yardim, The tunable fractional delay filter: passport to finegrain delay estimation, Invited Paper presented at *Fourth Cost 229 Workshop on Adaptive Methods and Emergent Techniques for Signal Processing and Communications*, pp. 9-24, Ljubljana, Slovenia, April 5-7, 1994. Also in *Electrotechnical Review*, Ljubljana, Slovenia, vol. 61, no. 4, pp. 232-242, 1994.

15. G. D. Cain, N. P. Murphy, and A. Tarczynski, Evaluation of several variable FIR fractional-sample delay filters, in *Proc. IEEE Int. Conf. Acoust. Speech Signal Processing* (ICASSP-94), vol. 3, pp. 621-624, Adelaide, Australia, April 19-22, 1994.

16. G. D. Cain, A. Yardim, and P. Henry, New FIR fractional delayors for effective signal interpolation, in *Proc. DSP-94 ESA Workshop on DSP for Space Applications*, pp. 42-49, London, Sept. 26-28, 1994.

17. G. D. Cain, A. Yardim, and P. Henry, Offset windowing for FIR fractional-sample delay, in *Proc. IEEE Int. Conf. Acoust. Speech and Signal Processing* (ICASSP-95), vol. 2, pp. 1276-1279, Detroit, Michigan, May 9-12, 1995.

18. T. Chen, H. P. Graf, and K. Wang, Speech-assisted video processing: interpolation and low-bitrate coding, in *Proc. 28th Annual Asilomar Conf. on Signals, Systems, and Computers*, pp. 975-979, Pacific Grove, California, Oct. 1994.

19. T. Chen, H. P. Graf, and K. Wang, Lip-synchronization using speech-assisted video processing, *IEEE Signal Processing Letters,* vol. 2, no. 4, pp. 57-59, April 1995.

20. T. Chen, personal communication with T. I. Laakso, May 10, 1995.

21. P. R. Cook, TBone: an interactive waveguide brass instrument synthesis workbench for the NeXT machine, in *Proc. Int. Computer Music Conf.* (ICMC-91), pp. 297-299, Montreal, Canada, Oct. 16-20, 1991.

22. P. R. Cook, A meta-wind-instrument physical model, and a meta-controller for real time performance control, in *Proc. Int. Computer Music Conf.* (ICMC-92), pp. 273-276, San Jose, California, Oct. 14-18, 1992.

23. 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.

24. R. E. Crochiere and L. R. Rabiner, *Multirate Digital Signal Processing.* Englewood Cliffs, New Jersey: Prentice-Hall, 1983.

25. 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 Processing* (ICASSP-91), vol. 5, pp. 3609-3612, Toronto, Canada, May 14-17, 1991.

26. A. G. Deczky, Equiripple and minimax (Chebyshev) approximations for recursive digital filters, *IEEE Trans. Acoust. Speech Signal Processing,* vol. 22, pp. 98-111, April 1974.

27. L. Erup, F. M. Gardner, and F. A. Harris, Interpolation in digital modems-part II: implementation and performance, *IEEE Trans. Comm.*, vol. 41, no. 6, pp. 998-1008, June 1993.

28. K.-P. Estola and T. Saramäki, A new method for designing equiripple error group delay filters, in *Proc. IEEE Int. Symp. Circuits Syst.* (ISCAS-85), vol. 1, pp. 271-274, Kyoto, Japan, June 5-7, 1985.

29. C. W. Farrow, A continuously variable digital delay element, in *Proc. IEEE Int. Symp. Circuits Syst.* (ISCAS-88), vol. 3, pp. 2641-2645, Espoo, Finland, June 6-9, 1988.

30. F. M. Gardner, Timing adjustment via interpolation in digital demodulators, European Space Agency Report, Contract 8022/88/NL/DG, Palo Alto, California, June 1990.

31. F. M. Gardner and L. Erup, Interpolation for timing adjustment in digital modems, in *Proc. Second Int. Workshop on Digital Signal Processing Techniques Applied to Space Communications* (DSP-90), ESA-WPP-019 1-2, Politecnico di Torino, Turin, Italy, Sept. 24-25, 1990.

32. F. M. Gardner, Interpolation in digital modems-part I: fundamentals, *IEEE Trans. Comm.*, vol. 41, no. 3, pp. 502-508, Mar. 1993.

33. M. C. Gill and L. P. Sabel, On the use of interpolation in digital demodulators, *Australian Telecommunications Review*, vol. 27, no. 2, pp. 25-31, 1993.

34. G. H. Golub and C. F. Van Loan, *Matrix Computations*. Second Edition. Baltimore, Maryland: Johns Hopkins University Press, 1989.

35. F. J. Harris, On the use of windows for harmonic analysis with the discrete Fourier transform, *Proc. IEEE*, vol. 66, no. 1, pp. 51-83, Jan. 1978.

36. S. Haykin, *Adaptive Filter Theory*. Englewood Cliffs, New Jersey: Prentice-Hall, 1986.

37. E. Hermanowicz, Explicit formulas for weighting coefficients of maximally flat tunable FIR delayers, *Electronics Letters*, vol. 28, no. 20, pp. 1936-1937, 24th Sept. 1992.

38. M. Ikehara, M. Funaishi, and H. Kuroda, Design of all-pass networks using Remez algorithm, in *Proc. IEEE Int. Symp. Circuits Syst.* (ISCAS-91), pp. 364-367, Westin Plaza, Singapore, June 11-14, 1991.

39. D. A. Jaffe and J. O. Smith, Extensions of the Karplus-Strong plucked string algorithm, *Computer Music J.*, vol. 7, no. 2, pp. 56-69, 1983. Reprinted in C. Roads (ed.), *The Music Machine*. Cambridge, Massachusetts: MIT Press, 1989, pp. 481-494.

40. Z. Jing, A new method for digital allpass filter design, *IEEE Trans. Acoust. Speech Signal Processing,* vol. 35, pp. 1557-1564, Nov. 1987.

41. L. J. Karam and J. H. McClellan, Complex Chebyshev approximation for FIR filter design, *IEEE Trans. Circuits Syst.-II: Analog and Digital Signal Processing,* vol. 42, no. 3, pp. 207-216, Mar. 1995.

42. 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 Processing* (ICASSP-91), vol. 5, pp. 3653-3656, Toronto, Canada, May 14-17, 1991.

43. M. Karjalainen and V. Välimäki, Model-based analysis/synthesis of the acoustic guitar, in *Proc. Stockholm Music Acoustics Conf. (SMAC 93)*, pp. 443-447, Stockholm, Sweden, July 28-Aug. 1, 1993.

44. M. Karjalainen, V. Välimäki, and Z. Jánosy, Towards high-quality synthesis of the guitar and string instruments, in *Proc. Int. Computer Music Conf.* (ICMC-93), pp. 56-63, Tokyo, Japan, Sept. 10-15, 1993.

45. K. Karplus and A. Strong, Digital synthesis of plucked-string and drum timbres, *Computer Music J.*, vol. 7, no. 2, pp. 43-55, 1983. Reprinted in C. Roads (ed.), *The Music Machine*. Cambridge, Massachusetts: MIT Press, 1989, pp. 467-479.

46. S. Kay, Some results in linear interpolation theory, *IEEE Trans. Acoust. Speech Signal Processing*, vol. 31, pp. 746-749, June 1983.

47. P. Knutson, D. McNeely, and K. Horlander, An optimal approach to digital raster mapper design, *IEEE Trans. Consumer Electronics*, vol. 37, no. 4, pp. 746-752, Nov. 1991.

48. C. C. Ko and Y. C. Lim, Approximation of a variable-length delay line by using tapped delay line processing, *Signal Processing,* vol. 14, no. 4, pp. 363-369, June 1988.

49. P. Kootsookos and R. C. Williamson, FIR approximation of fractional sample delay systems, *IEEE Trans. Circuits Syst.-II: Analog and Digital Signal Processing*, vol. 43, no. 2, Feb. 1996.

50. P. Kroon and B. S. Atal, Pitch prediction with high temporal resolution, in *Proc. IEEE Int. Conf. Acoust. Speech Signal Processing* (ICASSP-90), vol. 2, pp. 661-664, Albuquerque, New Mexico, April 3-6, 1990.

51. P. Kroon and B. S. Atal, On the use of pitch predictors with high temporal resolution, *IEEE Trans. Signal Processing*, vol. 39, no. 3, pp. 733-735, Mar. 1991.

52. T. I. Laakso, V. Välimäki, M. Karjalainen, and U. K. Laine, Real-time implementation techniques for a continuously variable digital delay in modeling musical instruments, in *Proc. Int. Computer Music Conf.* (ICMC-92), pp. 140-141, San Jose, California, Oct. 14-18, 1992.

53. T. I. Laakso, T. Q. Nguyen, and D. Koilpillai, Designing allpass filters using the eigenfilter method, in *Proc. IEEE Int. Conf. Acoust. Speech Signal Processing* (ICASSP-93), vol. 3, pp. 77-80, Minneapolis, Minnesota, April 27-30, 1993.

54. T. I. Laakso, S. J. Ovaska, Optimal polynomial predictors with application specific fixed prefilters, in *Proc. IEEE Int. Symp. Circuits Syst.* (ISCAS-93), vol. 1, pp. 351-354, Chicago, Illinois, May 3-6, 1993.

55. 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 Processing* (ICASSP-95), vol. 2, pp. 1304-1307, Detroit, Michigan, May 9-12, 1995.

56. 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 Symposium on Circuits and Systems* (MWSCAS-95), Rio de Janeiro, Brazil, Aug. 13-16, 1995.

57. R. Lagadec, D. Pelloni, and D. Weiss, A 2-channel, 16-bit digital sampling frequency converter for professional digital audio, in *Proc. IEEE Int. Conf. Acoust. Speech Signal Processing* (ICASSP-82), vol. 1, pp. 93-96, Paris, France, May 3-5, 1982.

58. U. K. Laine, Digital modelling of a variable length acoustic tube, in *Proc. Nordic Acoustical Meeting* (NAM-88), pp. 165-168, Tampere, Finland, June 15-17, 1988.

59. M. Lang, Optimal weighted phase equalization according to the L norm, *Signal Processing*, vol. 27, no. 1, pp. 87-98, April 1992.

60. M. Lang and T. Laakso, Design of allpass filters for phase approximation and equalization using LSEE error criterion, in *Proc. IEEE Int. Symp. Circuits Syst.* (ISCAS-92), pp. 2417-2420, San Diego, California, May 10-13, 1992.

61. M. Lang and J. Bamberger, Nonlinear phase FIR filter design with minimum LS error and additional constraints, in *Proc. IEEE Int. Conf. Acoust. Speech Signal Processing* (ICASSP-93), vol. 3, pp. 57-60. Minneapolis, Minnesota, April 27-30, 1993.

62. M. Lang, Beiträge zur Phasenapproximation mit Allpässen (Contributions to phase approximation by allpass filters, in German). Dr. Tech. Thesis, Erlangen, Germany: Lehrstuhl für Nachrichtentechnik, University of Erlangen, 1993.

63. M. Lang and T. I. Laakso, Simple and robust method for the design of allpass filters using least-squares phase error criterion, IEEE Trans. Circuits Syst.-II: Analog and Digital Signal Processing, vol. 41, no. 1, pp. 40-48, Jan. 1994.

64. M. Lang, personal communication with T. I. Laakso, Aug. 1995.

65. Y. C. Lim, J.-H. Lee, C. K. Chan, and R.-H. Yang, A weighted least-squares algorithm for quasi-equiripple FIR and IIR digital filter design, *IEEE Trans. Signal Processing*, vol. 40, no. 3, pp. 551-558, Mar. 1992.

66. G.-S. Liu and C.-H. Wei, Programmable fractional sample delay filter with Lagrange interpolation, *Electronics Letters*, vol. 26, no. 19, pp. 1608-1610, 13th Sept. 1990.

67. G.-S. Liu and C.-H. Wei, A new variable fractional sample delay filter with nonlinear interpolation, *IEEE Trans. Circuits Syst.-II: Analog and Digital Signal Processing*, vol. 39, no. 2, pp. 123-126, Feb. 1992.

68. 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 Communications and Technology (EUROSPEECH-89)*, vol. 2, pp. 509-512, Paris, France, September 1989. Also: 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 Processing (ICASSP-90)*, vol. 2, pp. 665-668, Albuquerque, New Mexico, April 3-6, 1990.

69. *The MATLAB User's Guide*. South Natick, Massachusetts: The Math-Works, Inc., 1989.

70. Y. Medan, Using super resolution pitch in waveform speech coders, in *Proc. IEEE Int. Conf. Acoust. Speech Signal Processing (ICASSP-91)*, vol. 1, pp. 633-636, Toronto, Canada, May 2-5, 1991.

71. G. A. Merchant and T. W. Parks, Efficient solution of Toeplitz-plus-Hankel coefficient matrix system of equations, *IEEE Trans. Acoust. Speech Signal Processing*, vol. 30, pp. 40-44, Feb. 1982.

72. H. Meyr and G. Ascheid, Synchronization in Digital Communications, vol. 2, to be published.

73. S. Minocha, S. C. Dutta Roy, and B. Kumar, A note on the FIR approximation of a fractional sample delay, *Int. J. Circuit Theory and Appl.*, vol. 21, no. 3, pp. 265-274, May-June 1993.

74. F. Mintzer, On half-band, third-band, and Nth-band FIR filters and their design, *IEEE Trans. Acoust. Speech Signal Processing*, vol. 30, pp. 734-738, Oct. 1982.

75. N. P. Murphy, A. Krukowski, and I. Kale, Implementation of wideband integer and fractional delay element, *Electronics Letters*, vol. 30, no. 20, pp. 1658-1659, 29th Sept. 1994.

76. T. Q. Nguyen, The design of arbitrary FIR digital filters using the eigenfilter method, *IEEE Trans. Signal Processing*, vol. 41, no. 3, pp. 1128-1139, Mar. 1993.

77. 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 Processing*, vol. 42, no. 9, pp. 2257-2263, Sept. 1994.

78. C. O'Neill, B. Murray, and A. D. Fagan, An efficient algorithm for pitch prediction using fractional delays, in *Proc. Sixth European Signal Processing Conf.* (EUSIPCO-92), vol. 1, pp. 319-322, Brussels, Belgium, Aug. 24-27, 1992.

79. G. Oetken, T. W. Parks, and H. W. Schüβler, New results in the design of digital interpolators, *IEEE Trans. Acoust. Speech Signal Processing,* vol. 23, no. 3, pp. 301-309, June 1975.

80. G. Oetken, A new approach for the design of digital interpolating filters, *IEEE Trans. Acoust. Speech Signal Processing,* vol. 27, no. 6, pp. 637-643, Dec. 1979.

81. A. Papoulis, *Signal Analysis*. New York: McGraw-Hill, 1977.

82. 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 Processing* (ICASSP-91), vol. 5, pp. 3613-3616, Toronto, Canada, May 14-17, 1991.

83. T. W. Parks and J. H. McClellan, A program for the design of linear phase finite impulse response digital filters, *IEEE Trans. Audio Electroacoust.*, vol. AU-20, pp. 195-199, Aug. 1972.

84. T. W. Parks and C. S. Burrus, *Digital Filter Design*. New York: John Wiley & Sons, 1987.

85. S.-C. Pei and J.-J. Shyu, Eigen-approach for designing FIR filters and allpass phase equalizers with prescribed magnitude and phase response, *IEEE Trans. Circuits Syst.-II: Analog and Digital Signal Processing*, vol. 39, no. 3, pp. 137-146, Mar. 1992.

86. L. E. Pellon, A double Nyquist digital product detector for quadrature sampling, IEEE Trans. Signal Processing, vol. 40, no. 7, pp. 1670-1681, July 1992.

87. K. Preuss, On the design of FIR filters by complex approximation, *IEEE Trans. Acoust. Speech Signal Processing*, vol. 37, pp. 702-712, May 1989.

88. 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 Processing* (ICASSP-87), vol. 2, pp. 896-899, Dallas, Texas, April 6-9, 1987.

89. R. P. Ramachandran and P. Kabal, Pitch prediction filters in speech coding, *IEEE Trans. Acoust. Speech and Signal Processing*, vol. 37, no. 4, pp. 467-478, April 1989.

90. V. Ramachandran, C. S. Gargour, and M. Ahmadi, Generation of digital transfer functions of the FIR type approximating $z^{-p/q}$, in *Proc. IEEE Int. Symp. Circuits Syst.* (ISCAS-90), vol. 4, pp. 3260-3262, New Orleans, Louisiana, May 1-3, 1990.

91. T. A. Ramstad, Sample-rate conversion by arbitrary ratios, in *Proc. IEEE Int. Conf. Acoust. Speech Signal Processing* (ICASSP-82), vol. 1, pp. 101-104, Paris, France, May 3-5, 1982.

92. T. A. Ramstad, Digital methods for conversion between arbitrary sampling frequencies, *IEEE Trans. Acoust. Speech Signal Processing*, vol. 32, no. 3, pp. 577-591, June 1984.

93. P. A. Regalia, Special filter designs, in S. K. Mitra and J. F. Kaiser (eds.), *Handbook for Digital Signal Processing.* New York: John Wiley & Sons, 1993, pp. 948-953.

94. M. Renfors and T. Saramäki, A class of approximately linear phase digital filters composed of allpass subfilters, in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS-86)*, pp. 678-681, San Jose, California, May 1986.

95. L. P. Sabel, A. Yardim, and G. D. Cain, An analysis of matched filter interpolator performance for digital demodulators, presented at *Third UK/Australian Int. Symp. on DSP for Communication Systems,* University of Warwick, Coventry, U.K., Dec. 12-14, 1994.

96. L. P. Sabel, G. D. Cain, A. Yardim, and T. Laakso, Effects of delay root raised-cosine filters on the performance of digital demodulators, in *Proc. Int. Conf. on Telecommunications* (ICT'95), pp. 75-79, Bali, Indonesia, April 3-5, 1995.

97. 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.

98. M. Schulist, Improvements of a complex FIR filter design algorithm, *Signal Processing,* vol. 20, no. 1, pp. 81-90, May 1990.

99. H. W. Schüβler, Digitale Signalverarbeitung II, Skriptum Nr. 4 (Lecture notes on DSP II no. 4, in German). Erlangen, Germany: Lehrstuhl für Nachrichtentechnik, Universität Erlangen-Nürnberg, 1984.

100. H. W. Schüβler and J. Weith, On the design of recursive Hilbert transformers, in *Proc. IEEE Int. Conf. Acoust. Speech Signal Processing* (ICASSP-87), vol. 2, pp. 876-879, Dallas, Texas, April 6-9, 1987.

101. S. Sivanand, J. F. Yang, and M. Kaveh, Focusing filters for wide-band direction finding, *IEEE Trans. Signal Processing,* vol. 39, no. 2, pp. 437-445, Feb. 1991.

102. S. Sivanand, Variable-delay implementation using digital FIR filters, in *Proc. Int. Conf. Signal Processing Appl. and Tech. (ICSPAT-92)*, pp. 1238-1243, Boston, Massachusetts, Nov. 2-5, 1992.

103. J. O. Smith and P. Gossett, A flexible sampling-rate conversion method, in *Proc. IEEE Int. Conf. Acoust. Speech Signal Processing (ICASSP-84)*, vol. 2, pp. 19.4.1-19.4.4, San Diego, California, Mar. 19-21, 1984.

104. J. O. Smith and B. Friedlander, Adaptive interpolated time-delay estimation, *IEEE Trans. Aerospace and Electronic Systems,* vol. AES-21, no. 3, pp. 180-199, Mar. 1985.

105. J. O. Smith, J. O. Smith III comments on Sullivan Karplus-Strong article, *Computer Music Journal*, vol. 15, no. 2, pp. 10-11, 1991.

106. J. O. Smith, Bandlimited interpolation—introduction and algorithm, unpublished manuscript, Nov. 5, 1992.

107. P. Steffen, On digital smoothing filters: a brief review of closed form solutions and two new approaches, *Circuits, Systems, and Signal Processing,* vol. 5, no. 2, pp. 187-210, 1986.

108. 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.

109. R. Sudhakar, R. C. Agarwal, and S. C. Dutta Roy, Time domain interpolation using differentiators, *IEEE Trans. Acoust. Speech and Signal Processing*, vol. 30, no. 6, pp. 992-997, Dec. 1982.

(continued on page 60)

Appendix A. Examples of FD FIR Filters

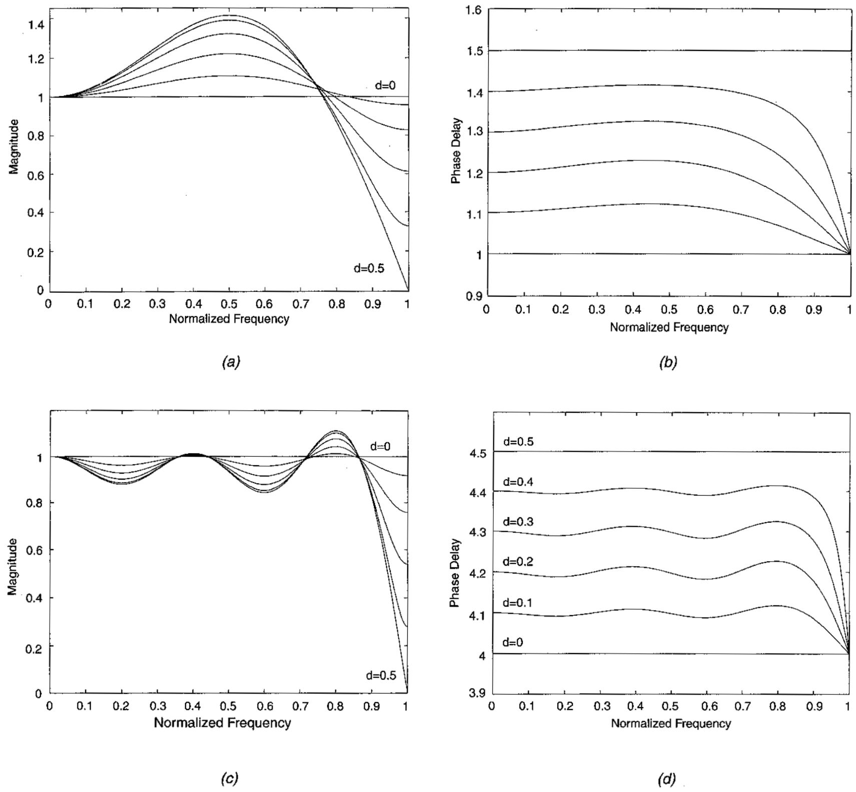


Fig. A1. Truncated LS FIR design. $L = 4$: (a) Magnitude and (b) phase delay response. $L = 10$: (c) Magnitude and (d) phase delay response.

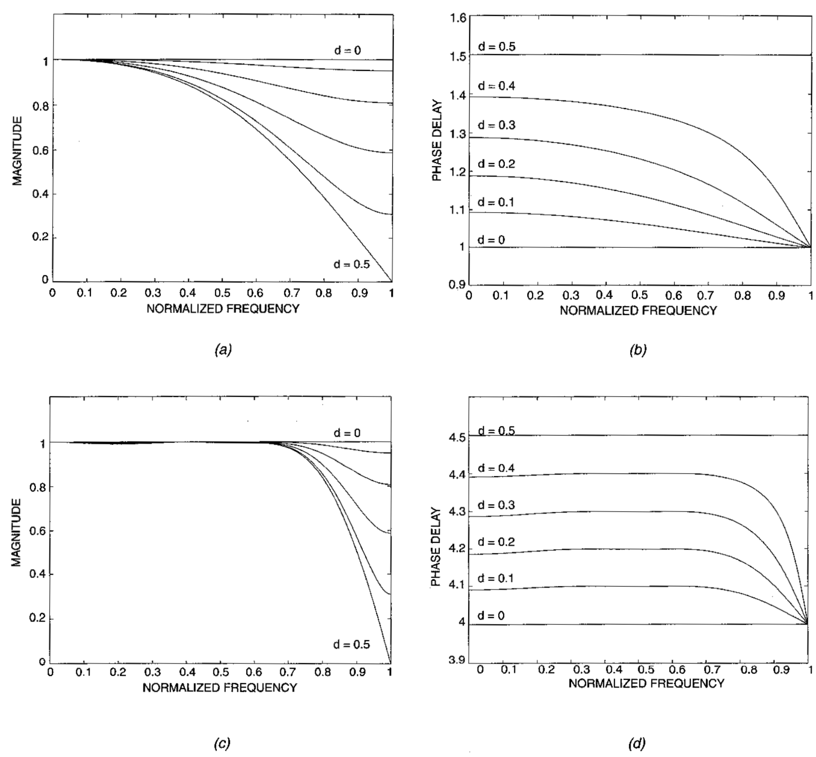


Fig. A2. Windowed LS FIR design. Fractionally shifted Dolph-Chebyshev window with 40 dB sidelobe ripple. $L = 4$: (a) Magnitude and (b) phase delay response. $L = 10$: (c) Magnitude and (d) phase delay response.

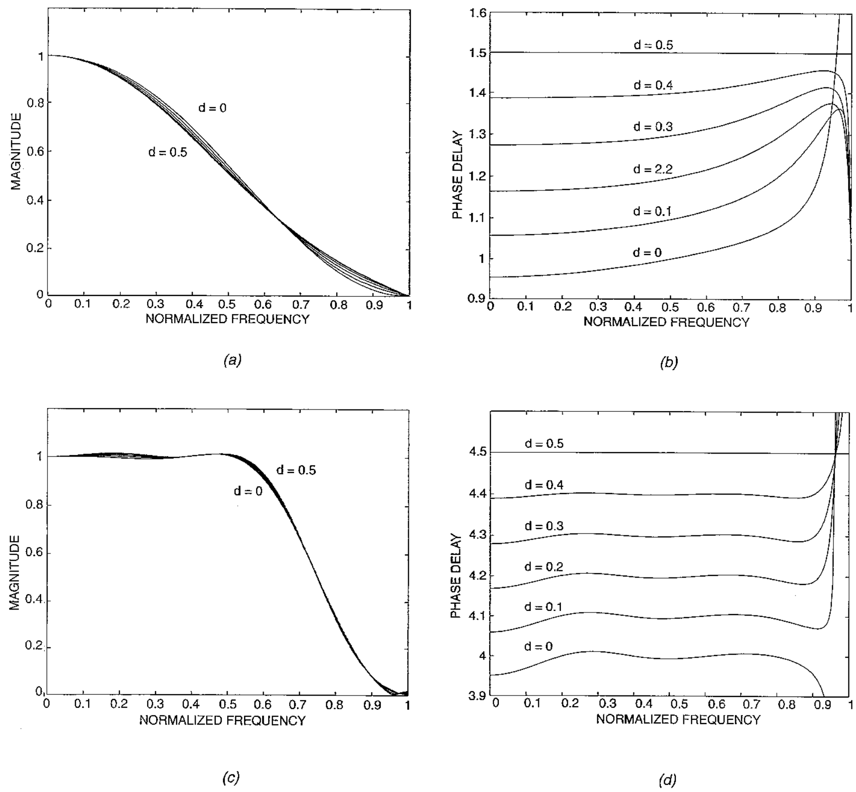


Fig. A3. LS FIR design with spline transition band ($P = 2$, $\omega_s = \pi$). $L = 4$, $\omega_p = 0.1\pi$: (a) Magnitude and (b) phase delay response. $L = 10$, $\omega_p = 0.8\pi$: (c) Magnitude and (d) phase delay response.

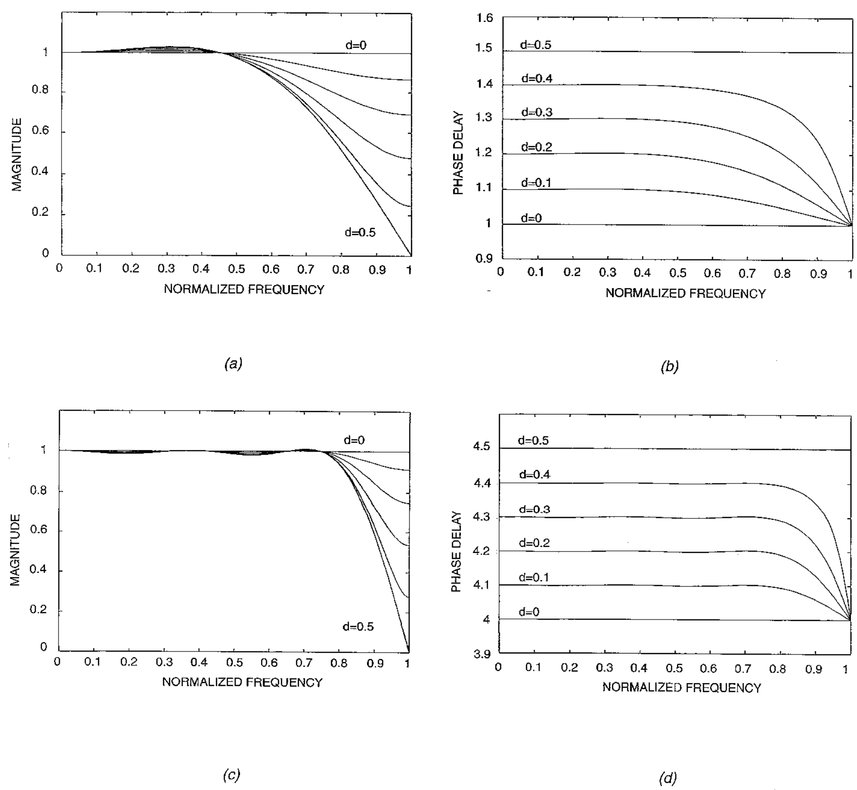


Fig. A4. General LS FIR design (ω_s=π). $L = 4$, $\alpha = 0.5$: (a) Magnitude and (b) phase delay response. $L = 10$, $\alpha = 0.8$: (c) Magnitude and (d) phase delay response.

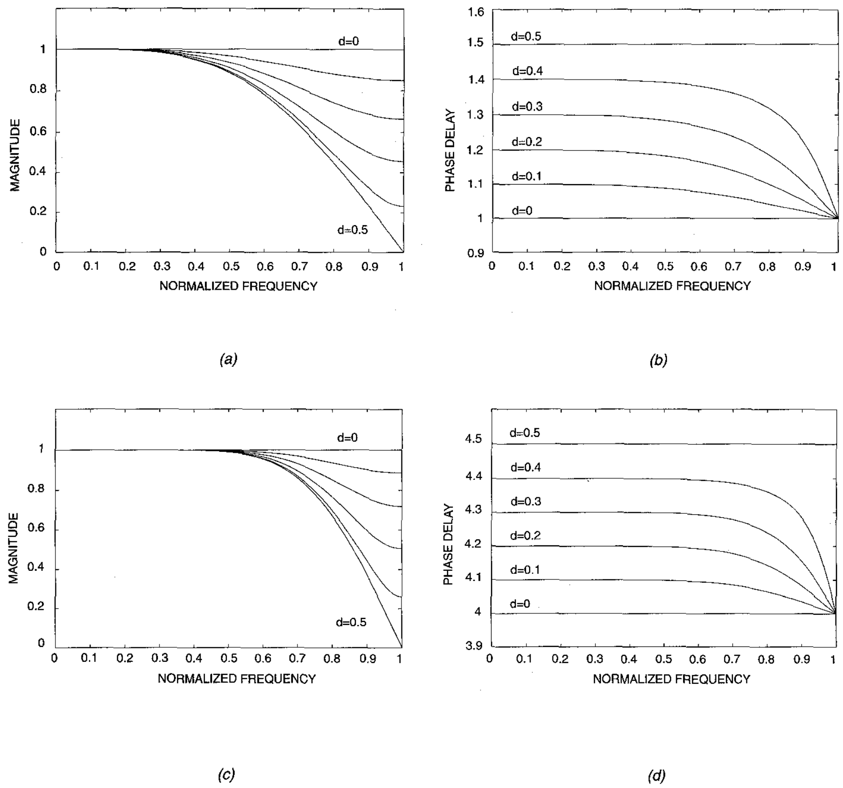


Fig. A5. Maximally flat FIR design (Lagrange interpolation). $L = 4$: (a) Magnitude and (b) phase delay response. $L = 10$: (c) Magnitude and (d) phase delay response.

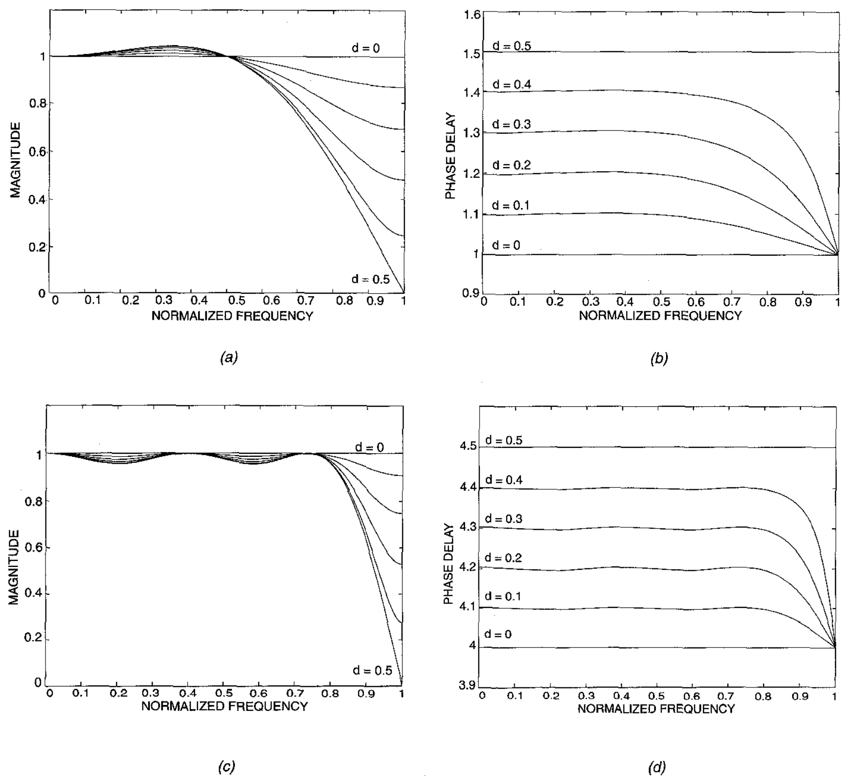


Fig. A6. Equiripple FIR design. $L = 4$, $\omega_p = 0.5\pi$: (a) Magnitude and (b) phase delay response. $L = 10$, $\omega_p = 0.8\pi$: (c) Magnitude and (d) phase delay response.

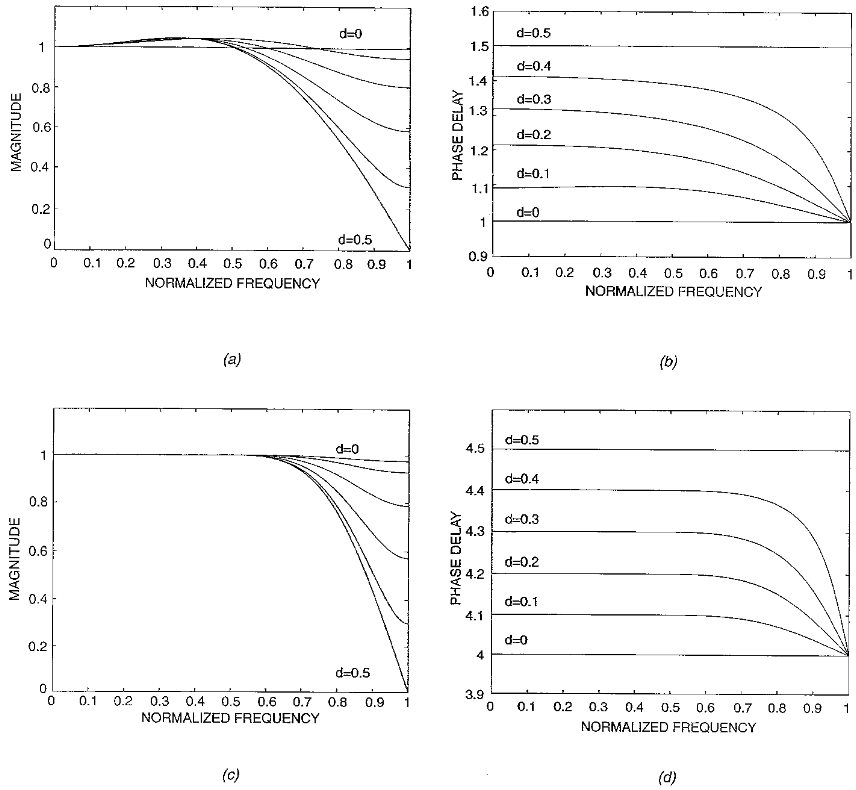


Fig. A7. Upsampling interpolation FIR design. $L = 4$, $\omega_p = 0.5\pi$: (a) Magnitude and (b) phase delay response. $L = 10$, $\omega_p = 0.8\pi$: (c) Magnitude and (d) phase delay response.

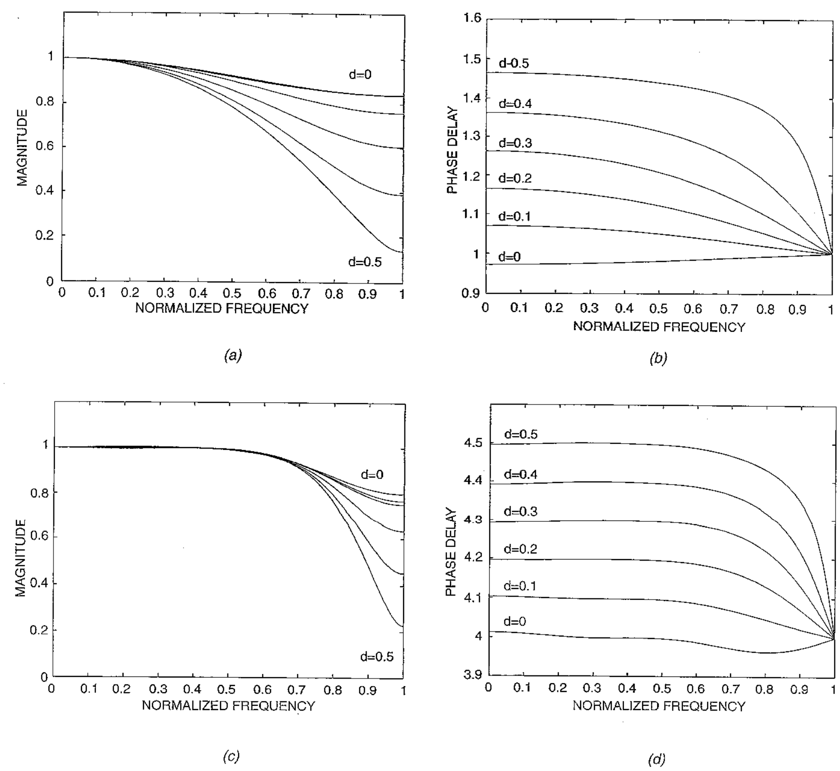


Fig. A8. The Fourier transform based control of the equiripple FIR design of Fig. A6. Fractionally shifted Dolph-Chebyshev window with 40 dB sidelobe ripple. $L = 4$: (a) Magnitude and (b) phase delay response.$L = 10$: (c) Magnitude and (d) phase delay response.

Appendix B. Examples of FD Allpass Filters

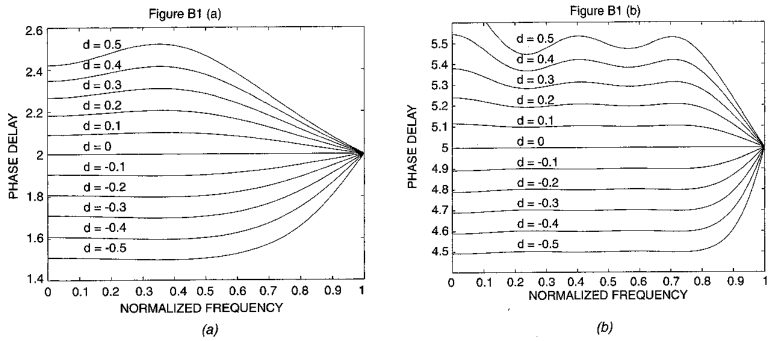


Fig. B1. Noniterative LS phase allpass design. (a) $N = 2$, $\omega_p = 0.5\pi$: phase delay response. (b) $N = 5$, $\omega_p = 0.8\pi$: phase delay response.

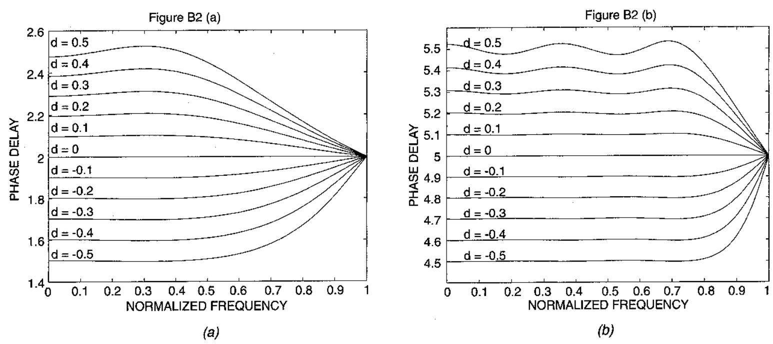


Fig. B2. Noniterative LS phase delay allpass design. (a) $N = 2$, $\omega_p = 0.5\pi$: phase delay response. (b) $N = 5$, $\omega_p = 0.8\pi$: phase delay response.

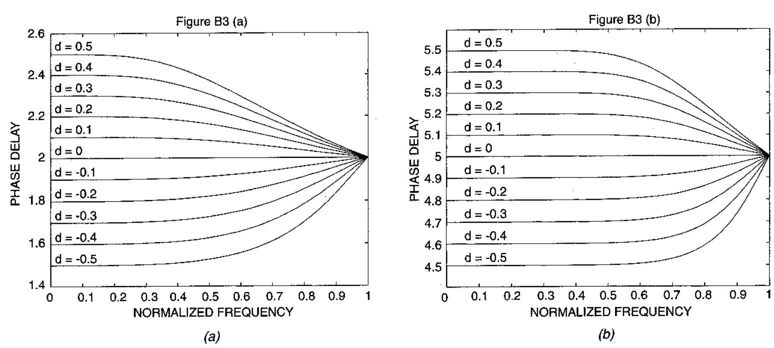


Fig. B3. Maximally flat group delay allpass design. (a) $N = 2$: phase delay response. (b) $N = 5$: phase delay response.

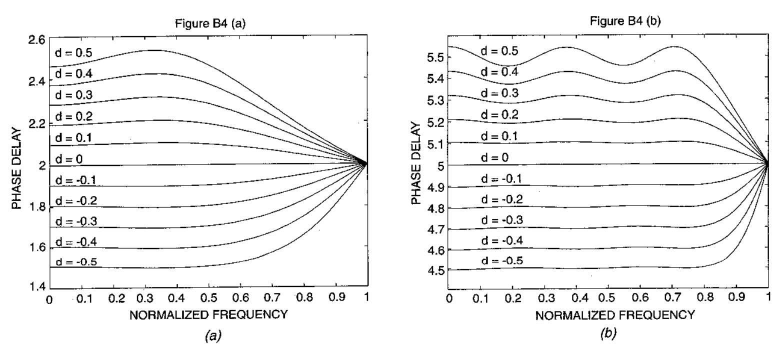


Fig. B4. Equiripple phase delay allpass design. (a) $N = 2$, $\omega_p = 0.5\pi$: phase delay response. (b) $N = 5$, $\omega_p = 0.8\pi$: phase delay response.

Appendix C. Minimizing the Quadratic Form $\mathbf{a}^T\mathbf{P}\mathbf{a}$

When the **P** matrix is positive definite, there exists a unique solution for a that minimizes $\mathbf{a}^T\mathbf{P}\mathbf{a}$ (provided that **a** is constrained to be nonzero, e.g., by setting $a_0 = 1$ as here). This solution can be found by eigenfilter techniques [116] or, alternatively, by solving the set of N equations in the conventional way. To express this solution in explicit form, we first partition the quadratic form as

$$\mathbf{a}^T\mathbf{P}\mathbf{a} = [1 \quad \mathbf{a}_1^T]\begin{bmatrix} p_0 & \mathbf{p}_1^T \\ \mathbf{p}_1 & \mathbf{P}_1 \end{bmatrix}\begin{bmatrix} 1 \\ \mathbf{a}_1 \end{bmatrix} = \mathbf{a}_1^T\mathbf{P}_1\mathbf{a}_1 + 2\mathbf{p}_1^T\mathbf{a}_1 + p_0 \tag{C1}$$

where $\mathbf{a}_1 = [a_1 \; a_2 \ldots \; a_N]^T$ is the vector of the free parameters to be solved. The optimal solution for $\mathbf{a}_1$ is found, e.g., by setting the derivative of (C1) with respect to $\mathbf{a}_1$ to zero, which yields the matrix equation

$$2\mathbf{P}_1\mathbf{a}_1 + 2\mathbf{p}_1 = 0 \tag{C2}$$

The formal solution is thus obtained as

$$\mathbf{a}_1 = -\mathbf{P}_1^{-1}\mathbf{p}_1 \tag{C3}$$

Commercial mathematical program packages like MATLAB [69] typically contain efficient and numerically robust routines for the solution of this kind of system of linear equations.

References *(continued from page 54)*

110. C. R. 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.

111. A. Tarczynski and G. D. Cain, Stabilisation of a class of feedback systems by means of fractional-sample delayors, *Applied Signal Processing*, vol. 1, no. 1, pp. 55-58, 1994.

112. A. Tarczynski, W. Kozinski, and G. D. Cain, Sampling rate conversion using fractional-sample delay, in *Proc. IEEE Int. Conf. Acoust. Speech Signal Processing (ICASSP-94),* vol. 3, pp. 285-288, Adelaide, Australia, April 19-22, 1994.

113. A. Tarczynski and G. D. Cain, Design of IIR fractional-sample delay filters, in *Proc. Second Int. Symp. DSP for Communications Systems (DSPCS-94)*, pages not numbered, Adelaide, Australia, April 26-29, 1994.

114. J.-P. Thiran, Recursive digital filters with maximally flat group delay, *IEEE Trans. Circuit Theory*, vol. CT-18, no. 6, pp. 659-664, Nov. 1971.

115. J.-P. Thiran, Equal-ripple delay recursive digital filters, *IEEE Trans. Circuit Theory,* vol. CT-18, no. 6, pp. 664-669, Nov. 1971.

116. P. P. Vaidyanathan and T. Q. Nguyen, Eigenfilters: a new approach to least-squares FIR filter design and applications including Nyquist filters, *IEEE Trans. Circuits Syst.*, vol. 34, no. 1, pp. 11-23, Jan. 1987.

117. P. P. Vaidyanathan and T. Q. Nguyen, A TRICK for the design of FIR half-band filters, *IEEE Trans. Circuits Syst.*, vol. 34, no. 3, pp. 297-300, Mar. 1987.

118. P. P. Vaidyanathan, Multirate digital filters, filter banks, polyphase networks, and applications: a tutorial, *Proc. IEEE,* vol. 78, no. 1, pp. 56-93, Jan. 1990.

119. P. P. Vaidyanathan, *Multirate Systems and Filter Banks.* Englewood Cliffs, New Jersey: Prentice-Hall, 1993.

120. V. Välimäki, M. Karjalainen, Z. Jánosy, and U. K. Laine, A real-time DSP implementation of a flute model, in *Proc. IEEE Int. Conf. Acoust. Speech Signal Processing (ICASSP-92)*, vol. 2, pp. 249-252, San Francisco, California, Mar. 23-26, 1992.

121. V. Välimäki, T. I. Laakso, M. Karjalainen, and U. K. Laine, A new computational model for the clarinet, in *Proc. Int. Computer Music Conf. (ICMC-92)*, Addendum (pages not numbered), San Jose, California, Oct. 14-18, 1992.

122. V. Välimäki, T. I. Laakso, and M. Karjalainen, Fractional delay digital filters, in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS-93)*, vol. 1, pp. 355-358, Chicago, Illinois, May 3-6, 1993.

123. V. Välimäki, M. Karjalainen, and T. I. Laakso, Modeling of woodwind bores with finger holes, in *Proc. Int. Computer Music Conf.* (ICMC-93), pp. 32-39, Tokyo, Japan, Sept. 10-15, 1993.

124. V. Vèlimèki, M. Karjalainen, and T. Kuisma, Articulatory control of a vocal tract model based on fractional delay waveguide filters, in *Proc. IEEE Int. Symp. Speech, Image Processing and Neural Networks (ISSIPNN-94)*, vol. 2, pp. 571-574, Hong Kong, April 13-16, 1994.

125. V. Välimäki, M. Karjalainen, and T. Kuisma, Articulatory speech synthesis based on fractional delay waveguide filters, in *Proc. IEEE Int. Conf. Acoust. Speech Signal Processing (ICASSP-94)*, vol. 1, pp. 585-588, Adelaide, Australia, April 19-22, 1994.

126. V. Välimäki, *Fractional Delay Waveguide Modeling of Acoustic Tubes.* Report no. 34, Espoo, Finland: Helsinki Univ. of Tech., Faculty of Electrical Eng., Laboratory of Acoustics and Audio Signal Processing, 133 p., July 1994.

127. V. Välimäki and M. Karjalainen, Digital waveguide modeling of wind instrument bores constructed of truncated cones, in *Proc. Int. Computer Music Conf.* (ICMC+94), pp. 423-430, Aarhus, Denmark, Sept. 12-17, 1994.

128. V. Välimäki and M. Karjalainen, Improving the Kelly-Lochbaum vocal tract model using conical tube sections and fractional delay filtering techniques, in *Proc. Int. Conf. Spoken Language Processing (ICSLP-94)*, vol. 2, pp. 611-614, Yokohama, Japan, Sept. 18-22, 1994.

129. V. Välimäki, J. Huopaniemi, M. Karjalainen, and Z. Jçnosy, Physical modeling of plucked string instruments with application to real-time sound synthesis, presented at the *AES 98th Convention*, preprint no. 3956, Paris, France, Feb. 25-28, 1995.

130. V. Välimäki, A new filter implementation strategy for Lagrange interpolation, in *Proc. IEEE Int. Symp. Circuits and Systems (ISCAS-95)*, vol. 1, pp. 361-364, Seattle, Washington, April 29-May 3, 1995.

131. V. Välimäki and M. Karjalainen, Implementation of fractional delay waveguide models using allpass filters, in *Proc. IEEE Int. Conf. Acoust. Speech Signal Processing (ICASSP-95)*, vol. 2, pp. 1524-1527, Detroit, Michigan, May 9-12, 1995.

132. V. Välimäki, T. I. Laakso, and J. Mackenzie, Time-varying fractional delay filters, in *Proc. Finnish Signal Processing Symp. (FINSIG-95)*, pp. 118-122, Espoo, Finland, June 2, 1995.

133. 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. (ICMC-95)*, pp. 327-334, Banff, Canada, Sept. 3-7, 1995.

134. V. Vèlimèki and T. I. Laakso, Elimination of transients in tunable recursive digital filters, to be published, Oct. 1995.

135. V. Välimäki, *Discrete-Time Modeling of Acoustic Tubes Using Fractional Delay Filters.* Dr. Tech. Thesis, Espoo, Finland: Helsinki Univ. of Tech., Faculty of Electrical Eng., Laboratory of Acoustics and Audio Signal Processing, Dec. 1995.

136. V. Zivojnovic and H. Meyr, Design of optimum interpolation filters for digital demodulators, in *Proc. IEEE Int. Symp. Circuits and Systems (ISCAS-93)*, vol. 1, pp. 140-143, Chicago, Illinois, May 3-6, 1993.

137. U. Zölzer and T. Boltze, Interpolation algorithms: theory and application, presented at the *AES 97th Convention*, preprint no. 3898, San Francisco, California, Nov. 10-13, 1994.