On Fast FIR Filters Implemented as Tail-Canceling IIR Filters

Avery Wang and Julius O. Smith, III, *Member, IEEE*

***Abstract*—We have developed an algorithm based on synthetic division for deriving the transfer function that cancels the tail of a given arbitrary rational (IIR) transfer function after a desired number of time steps. Our method applies to transfer functions with repeated poles, whereas previous methods of tail-subtraction cannot. We use a parallel state-variable technique with periodic refreshing to induce finite memory in order to prevent accumulation of quantization error in cases where the given transfer function has unstable modes. We present two methods for designing linear-phase truncated IIR (TIIR) filters based on antiphase filters. We explore finite-register effects for unstable modes and provide bounds on the maximum TIIR filter length. In particular, we show that for unstable systems, the available dynamic range of the registers must be three times that of the data. Considerable computational savings over conventional FIR filters are attainable for a given specification of linear-phase filter. We provide examples of filter design. We show how to generate finite-length polynomial impulse responses using TIIR filters. We list some applications of TIIR filters, including uses in digital audio and an algorithm for efficiently implementing Kay's optimal high-resolution frequency estimator.**

I. Introduction

INFINITE impulse response (IIR) recursive linear digital filters are widely used because of their low computational cost and low storage overhead requirements. Finite impulse response (FIR) filters, on the other hand, allow the possibility of implementing linear-phase linear digital filters that have constant group delay across all frequencies. The tradeoff is that to achieve similar magnitude transfer functions, FIR filters usually require much larger filter orders than their IIR counterparts. For example, a general Nth-order FIR filter requires $N+1$ multiplies and N adds. In certain cases, however, FIR filters that have an operation count comparable with that of an IIR filter while maintaining the linear phase property may be designed.

The set of FIR's that may be efficiently implemented includes those that are truncated IIR (TIIR) sequences of low order. Although the following results were derived independently of Saramäki and Fam [1], [3], the idea of using truncated IIR filters to generate linear-phase filters was originally introduced by Fam [3]. Fam and Saramäki [1], [3] deal with unstable hidden modes due to pole-zero cancellations outside of the unit circle by employing a switching-and-resetting algorithm to reduce the effects of quantization error buildup. We introduce a slightly more efficient version of this idea, as well as an error analysis.

In this paper, we describe an algorithm for the efficient implementation of certain classes of FIR filters. We introduce an extension of the TIIR algorithm that allows the truncation of arbitrary IIR filter tails and provides a way to implement polynomial impulse responses. Additionally, we present an analysis of the effects of limited numerical precision and provide design guidelines for designing systems with acceptable noise tolerance.

Manuscript received April 29, 1994; revised November 10, 1995. This work was supported in part by a National Science Foundation Graduate Fellowship. The associate editor coordinating the review of this paper and approving it for publication was Prof. Demetris Lainiotis.

A. Wang is with Chromatic Research, Sunnyvale, CA 94089 USA (e-mail: avery@chromatic.com).

J. O. Smith is with the Center for Computer Research in Music and Acoustics, Stanford University, Stanford, CA 94305 USA (e-mail: jos@ccrma.stanford.edu).

Publisher Item Identifier S 1053-587X(97)04230-X.

II. Definitions

A general causal Nth-order FIR filter consists of a tapped delay line N elements long and a table of $N+1$ impulse response coefficients $\{h_0, \cdots, h_N\}$ such that at each time step n, the output

$$y[n] = \sum_{k=0}^{N} h_k x[n-k] \tag{1}$$

is formed. The transfer function has the simple form

$$H_{\mathrm{FIR}}(z) \triangleq h_0 + h_1 z^{-1} + \cdots + h_N z^{-N} \tag{2}$$

$$= z^{-N} C(z) \tag{3}$$

where $C(z)$ is the Nth-degree polynomial formed by the h_k. On the other hand, a causal Pth-order IIR filter has the relation

$$y[n] = -\sum_{k=1}^{P} a_k y[n-k] + \sum_{\ell=0}^{P} b_\ell x[n-\ell]. \tag{4}$$

The corresponding transfer function is

$$H_{\mathrm{IIR}}(z) \triangleq \frac{b_0 + b_1 z^{-1} + \cdots + b_P z^{-P}}{1 + a_1 z^{-1} + \cdots + a_P z^{-P}} \tag{5}$$

$$\triangleq \frac{B(z)}{A(z)} \tag{6}$$

$$\triangleq h_0 + h_1 z^{-1} + h_2 z^{-2} + \cdots \tag{7}$$

where both

$$A(z) = z^P + a_1 z^{P-1} + \cdots + a_P \tag{8}$$

and

$$B(z) = b_0 z^P + b_1 z^{P-1} + \cdots + b_P \tag{9}$$

may be assumed to be relatively prime Pth-degree polynomials in z, and $A(z)$ is monic by construction.

Group delay [4] is defined by

$$\tau_d(\omega) \triangleq -\frac{d \arg\{H(e^{j\omega})\}}{d\omega}. \tag{10}$$

The group delay at normalized frequency $\omega = 2\pi f/f_s$, where f_s is the sampling frequency, is the number of samples of delay experienced by the amplitude envelope of a narrow-band input signal centered at ω.

A *linear-phase filter* is one such that the phase response at a given frequency is a linear function of frequency, i.e., $\arg\{H(e^{j\omega})\} = K_1\omega + K_2$ for some constants K_1 and K_2. From this property, we see immediately that the group delay is constant for all frequencies. Filters with linear phase response are often desirable because they have no frequency-dependent temporal distortion. A stable IIR filter with nonzero poles cannot have linear phase. However, an FIR filter with coefficients $\{h_0, \cdots, h_N\}$ has linear phase if there exists a ψ such that for all $k \in \{0, \cdots, N\}$

$$h_{N-k} = e^{j\psi} h_k^* \tag{11}$$

i.e., if the reversed coefficients are the complex conjugates of the forward sequence plus a constant phase shift [5]. The group delay is then

$$\tau_d(f) = \frac{N}{2}. \tag{12}$$

III. Truncated IIR (TIIR) Filters

Consider an FIR filter having a truncated geometric sequence $\{h_0, h_0p, \cdots, h_0p^N\}$ as an impulse response. This filter has the same impulse response for the first $N+1$ terms as the one-pole IIR filter with transfer function

$$H_{\text{IIR}}(z) = \frac{h_0}{1 - pz^{-1}}. \tag{13}$$

If we subtract off the tail of the impulse response, we obtain

$$H_{\text{FIR}}(z) = h_0 + h_0pz^{-1} + \cdots + h_0p^Nz^{-N} \tag{14}$$

$$= h_0\frac{1 - p^{N+1}z^{-(N+1)}}{1 - pz^{-1}}. \tag{15}$$

The time-domain recursion for this filter is

$$y[n] = \sum_{k=0}^{N} h_0p^k x[n-k] \tag{16}$$

$$= py[n-1] + h_0\{x[n] - p^{N+1}x[n-(N+1)]\}. \tag{17}$$

We see that the first formulation (16) requires $N+1$ multiplies and N adds to implement directly, whereas the second formulation (17) requires only three multiplies and two adds, independent of N. Thus, we see that if we can represent an FIR sequence as a truncated exponential sequence, a tremendous savings in computation can be achieved. Note that the $x[n-(N+1)]$ term in (17) still requires a delay line to be maintained, and thus, there are no savings in storage.

Notice that there is a pole-zero cancellation in the representation given by (15). If $|p| < 1$, there is no problem since the system is inherently stable. If $|p| \geq 1$, however, then there is a potential problem due to the hidden mode. We will deal with this in Section IV, where we will see how to run TIIR filters with unstable modes.

The idea of this section was used by Fam [3], where a partial fraction expansion of a transfer function is taken, and each mode is truncated separately using (15). This method works only for cases in which the multiplicity of each pole is one so that the each mode exhibits a simple exponential decay.

A. Extension to Higher Order TIIR Sequences

We may extend the idea illustrated in the previous section for the one-pole case to computing the TIIR sequence of any rational $H(z)$. The general procedure is to find the "tail" IIR transfer function

$$H'_{\text{IIR}}(z) = h'_0z^{-1} + h'_1z^{-2} + \cdots \triangleq h_{N+1}z^{-1} + h_{N+2}z^{-2} + \cdots \tag{18}$$

whose impulse response, except for a time shift of N steps, matches the tail of the transfer function $H_{\text{IIR}}(z)$, which we would like to truncate after time step N.

We multiply (7) by z^N and obtain

$$z^NH_{\text{IIR}}(z) = h_0z^N + \cdots + h_{N-1}z + h_N + h_{N+1}z^{-1} + h_{N+2}z^{-2} + \cdots \tag{19}$$

$$= C(z) + H'_{\text{IIR}}(z) \tag{20}$$

$$= \frac{z^NB(z)}{A(z)} \tag{21}$$

$$= C(z) + \frac{B'(z)}{A(z)} \tag{22}$$

where $\text{Deg}\{B'(z)\} < \text{Deg}\{A(z)\} = P$. We may assume that $\text{Deg}\{B'(z)\} = P-1$. $B'(z)$ is unique and may be obtained by performing synthetic division on $z^NB(z)$ by $A(z)$ and finding the remainder. Thus, $z^NB(z) \equiv B'(z) \pmod{A(z)}$.

Once we have obtained $B'(z)$, we have $H'_{\text{IIR}}(z) = B'(z)/A(z)$, and we may write

$$H_{\text{FIR}}(z) = H_{\text{IIR}}(z) - z^{-N}H'_{\text{IIR}}(z) \tag{23}$$

$$= \frac{B(z) - z^{-N}B'(z)}{A(z)}. \tag{24}$$

The corresponding system is

$$y[n] = -\sum_{k=1}^{P} a_ky[n-k] + \sum_{\ell=0}^{P} b_\ell x[n-\ell] - \sum_{m=0}^{P-1} b'_m x[n-m-(N+1)] \tag{25}$$

using the representation in (24).

The fact that the denominators of the transfer functions $H_{\text{IIR}}(z)$ and $H'_{\text{IIR}}(z)$ are the same allows additional savings in computational cost due to the fact that the original IIR and tail IIR dynamics are the same and do not need to be performed twice. The term $z^{-(N+1)}z^{-(P-1)}B'(z)$ serves to zero out the dynamics at the end of the delay line and requires only an additional P multiplies and $P-1$ adds. The computational

cost of this general truncated Pth-order IIR system is $3P+1$ multiplies and $3P-2$ adds, independent of N. Thus, a net computational savings with this class of FIR filters is achieved if $N > 3P$.

The storage costs for this filter are P output samples for the IIR feedback dynamics, N input samples of the FIR filter, and an additional P input samples for the tail-cancellation dynamics, yielding $N+P$ input delay samples, of which only the first and last P are used, and P output delay samples. Thus, the fast FIR algorithm requires $2P$ more storage samples than a direct FIR implementation.

As in the previous section, we observe cautiously that the effect of subtracting the tail IIR response $z^{-N}H'_{\mathrm{IIR}}(z)$ from $H_{\mathrm{IIR}}(z)$ is to cancel all the poles in (24), which is to be expected since an FIR filter is an all-zero filter.

B. Other Architectures

The direct implementation specified by (25) may not be desirable for various reasons. For example, one may choose to use a factored structure such as the cascaded biquad or the parallel partial fraction form. The latter form is given as

$$H(z) = \sum_{k=1}^{N_p} \sum_{\ell=1}^{M_k} \frac{C_{k,\ell}}{(1-p_k z^{-1})^\ell} \tag{26}$$

where N_p is the number of distinct poles, and M_k is the multiplicity of the kth pole. The (k, ℓ) term of the partial fraction expansion corresponds to a filter with impulse response

$$h_{k,\ell}[n] = C_{k,\ell}\binom{n+\ell-1}{\ell-1}p_k^n. \tag{27}$$

To form the TIIR filter, a tail IIR filter is derived for each partial fraction using synthetic division as outlined in Section III-A. Each TIIR response is calculated separately, and the results are added together to form the complete response. The factorization need not be as complete as outlined in (26). One may choose an intermediate level of factorization, leaving some factors lumped together and others separated from each other. For example, one may want to group complex-conjugate pairs together to avoid complex arithmetic. Alternatively, one may wish to leave terms with the same poles together as in

$$H(z) = \sum_{k=1}^{N_p} \frac{B_k(z)}{(z-p_k)^{M_k}} \tag{28}$$

since calculating the tail IIR response for each nth-order multiplicity term yields a degree $n-1$ polynomial numerator anyway. The impulse response of the kth partial fraction in this case is

$$h_k[n] = \sum_{\ell=0}^{M_k} b_{k,\ell}\binom{n-\ell+M_k-1}{M_k-1}p_k^{n-\ell} \tag{29}$$

where the $b_{k,\ell}$, $\ell = 0, \cdots, M_k$ are the coefficients of $B_k(z)$.

Another example is to group together the stable factors, i.e., those with poles p_k such that $|p_k| < 1$, and implement separately the unstable poles for which $|p_k| > 1$. These strategies will become useful in Section V.

IV. Unstable Hidden Modes

We now address the issue of pole-zero cancellation and the resulting hidden modes in the fast FIR algorithm. Although the naïve Fast FIR algorithm of (25) works in theory, there is the practical matter of quantization error due to finite register lengths when dealing with truncated unstable IIR systems, i.e., those with poles p_k such that $|p_k| > 1$. Consider the system in (17). We may model quantization error as an independent, identically distributed (IID) noise signal $\nu[n]$ with zero mean and variance σ_ν^2 input to the system of (17).[1] The output equation is then

$$y[n] = py[n-1] + h_0\{x[n] - p^{N+1}x[n-(N+1)]\} + \nu[n] \tag{30}$$

$$= h_0 \sum_{k=0}^{N-1} p^k x[n-k] + \sum_{k=0}^{n} p^k \nu[n-k]. \tag{31}$$

Thus, the accumulated noise has zero mean, but its variance grows as

$$\sigma_{\mathrm{acc}}^2[n] = \sum_{k=0}^{n} |p|^{2k}\sigma_\nu^2 \tag{32}$$

$$= \sigma_\nu^2 \frac{1-|p|^{2(n+1)}}{1-|p|^2} \tag{33}$$

so that

$$\lim_{n\to\infty} \sigma_{\mathrm{acc}}^2[n] = \begin{cases} +\infty, & |p| \geq 1 \\ \dfrac{\sigma_\nu^2}{1-|p|^2}, & |p| < 1. \end{cases} \tag{34}$$

For a higher-order system, as described in Section III-A, this analysis gives an estimate of the noise accumulation behavior if p is the largest-magnitude pole since its dynamics dominate the behavior of the system. A direct implementation of an FIR system, such as in (16), does not have noise accumulation problems because of its finite memory. We trade computational cost for noise sensitivity in TIIR systems.

Recalling the discussion in Section III-B, we may factor the unstable modes of a system apart from the transfer function so that

$$H(z) = H^{\downarrow}(z) + H^{\uparrow}(z) \tag{35}$$

$$= \frac{B^{\downarrow}(z)}{A^{\downarrow}(z)} + \frac{B^{\uparrow}(z)}{A^{\uparrow}(z)} \tag{36}$$

where the "$\downarrow$" and "$\uparrow$" superscripts denote "stable hidden modes" and "unstable hidden modes," respectively. Thus, we may implement the two filters of orders $P^{\downarrow}$ and $P^{\uparrow}$, where $P^{\downarrow} + P^{\uparrow} = P$, and add together the outputs. $H^{\downarrow}(z)$ may be

[1] Finite-register effects may be modeled as the sum of q uniformly distributed IID $[-\epsilon, +\epsilon]$, where $\epsilon > 0$ is the smallest quantity such that $x + \epsilon \neq x$ in machine arithmetic, and q is the number of additive terms. If q is relatively large, the total noise may be modeled as a Gaussian distribution. For fixed-point arithmetic, ϵ is constant $\forall\, x$. Floating-point arithmetic presents difficulties since the effective ϵ varies proportionally with the magnitude of x. Assuming independence and fixed-point arithmetic, we may model quantization noise as a Gaussian random variable with variance $\sigma_\nu^2 = q\epsilon^2/3$. A thorough analysis of quantization noise statistics is beyond the scope of this paper.

TABLE I
FAST FIR ALGORITHM FOR AN UNSTABLE TRUNCATED IIR (TIIR) FILTER

$n \equiv 0 \pmod N$	$n \not\equiv 0 \pmod N$
Primary TIIR Filter	
$w_1[N+k] \leftarrow 0, \quad k = 1, \ldots, P-1$ $w_1[k] \leftarrow w_1[k-1], \quad k = N, \ldots, 1$ $w_1[0] \leftarrow x[n]$ $q_1[k] \leftarrow q_2[k-1], \quad k = P, \ldots, 1$ $q_1[0] \leftarrow -\sum_{k=1}^{P} a_k q_1[k] + \sum_{\ell=0}^{P} b_\ell w_1[\ell]$ $y[n] \leftarrow q_1[0]$	$w_1[k] \leftarrow w_1[k-1], \quad k = N+P, \ldots, 1$ $w_1[0] \leftarrow x[n]$ $q_1[k] \leftarrow q_1[k-1], \quad k = P, \ldots, 1$ $q_1[0] \leftarrow -\sum_{k=1}^{P} a_k q_1[k] + \sum_{\ell=0}^{P} b_\ell w_1[\ell] - \sum_{m=0}^{P-1} b'_m w_1[m+N+1]$ $y[n] \leftarrow q_1[0]$
Auxiliary TIIR Filter	
$w_2[k] \leftarrow 0, \quad k = 1, \ldots, P-1$ $w_2[0] \leftarrow x[n]$ $q_2[k] \leftarrow 0, \quad k = 1, \ldots, P-1$ $q_2[0] \leftarrow b_0 w_2[0]$	$w_2[k] \leftarrow w_2[k-1], \quad k = P, \ldots, 1$ $w_2[0] \leftarrow x[n]$ $q_2[k] \leftarrow q_2[k-1], \quad k = P, \ldots, 1$ $q_2[0] \leftarrow -\sum_{k=1}^{P} a_k q_2[k] + \sum_{\ell=0}^{P} b_\ell w_2[\ell]$

implemented as described in Section III-A, but $H^{\uparrow}(z)$ must be handled carefully.

Alternatively, we may factor the transfer function into stable and unstable parts

$$H(z) = H^{\downarrow}(z) H^{\uparrow}(z) \tag{37}$$

$$= \frac{B^{\downarrow}(z)}{A^{\downarrow}(z)} \frac{B^{\uparrow}(z)}{A^{\uparrow}(z)} \tag{38}$$

in which case, the stable and unstable filters are cascaded. Note that in this factorization, $B^{\downarrow}(z)$ and $B^{\uparrow}(z)$ are not uniquely determined.

A possible algorithm for stabilizing an unstable TIIR filter using two parallel sets of state variables is given in Table I. The algorithm cycles with a period of N time steps. For cycle number k, at time step $n = kN$, we clear the auxiliary filter's input delay line $w_2[\cdot]$ and vector of state variables $q_2[\cdot]$, each of length P. Otherwise, the primary and auxiliary systems evolve with the same dynamics. We have taken into account the fact that during the first N time steps, the tail canceling terms due to $z^{-N}B'(z)$ in the auxiliary filter are identically zero, and hence, those terms are not included. Since the FIR impulse response has length $N+1$, we see that at time $n = kN + N$, the two systems have identical output values, except for the amount of accumulated noise. The state variables $q_1[\cdot]$ and $q_2[\cdot]$ are not identical because the input histories are slightly different. However, since we are only concerned with the output, this does not matter. We then copy the auxiliary system's state variables to the primary system and repeat the cycle. This algorithm may be optimized by realizing that the values for $w_2[m]$ are zero for $m > N$ during each cycle. Thus, the last summation in the dynamics for the auxiliary system shown in Table I is omitted for the cases where $n \equiv 1, \cdots, N \pmod N$.

Under this scheme, the amount of time that quantization errors can accumulate varies from a minimum of N time steps to a maximum of $2N$, depending on when each error occurs. The minimum is due to the amount of time necessary to allow the auxiliary system to evolve to having the same output value as the primary system. For noise accumulation analysis in this system, we may assume that the unstable system dynamics are dominated by the mode with the largest eigenvalue $p_{\max}$ such that $|p_{\max}| > 1$. We assume for the present analysis that the multiplicity of $p_{\max}$ is one. Thus, we must reckon for tolerable performance with

$$\sigma^2_{\max} \geq \sigma^2_\nu \frac{1 - |p_{\max}|^{4N}}{1 - |p_{\max}|^2}. \tag{39}$$

If the maximum tolerable noise variance is σ^2_{tol}, then we must have $\sigma^2_{\max} < \sigma^2_{\text{tol}}$ so that, approximately

$$N < \frac{\log\left[1 + \dfrac{(|p_{\max}|^2 - 1)\sigma^2_{\text{tol}}}{\sigma^2_\nu}\right]}{4\log(|p_{\max}|)}. \tag{40}$$

Thus, given $p_{\max}$, σ^2_{tol} and quantization noise variance σ^2_ν, we have a fundamental limitation on the length of the effective impulse response. A refinement of this bound is given below in Section V-C.

The computational cost for implementing an unstable TIIR filter is, at most, twice the cost of a stable TIIR filter. The extra cost is due to the cost of implementing the auxiliary TIIR filter, which costs about $2P$ operations per sample. Thus, the computational cost of the unstable TIIR filter is approximately $5P$ multiplies per input sample. The number of adds is about the same. If we take into account the fact that the feedforward terms due to $B(z)$ are the same (except for cases where $w_2[k]$ has been set to zero) in both the primary and auxiliary systems, we may optimize by avoiding duplicate calculation of these

TABLE II
FAST FIR ALGORITHM FOR A REVERSED UNSTABLE TRUNCATED IIR (TIIR) FILTER DERIVED FROM A STABLE TIIR FILTER

$n \equiv 0 \pmod N$	$n \not\equiv 0 \pmod N$
Primary TIIR Filter	
$w_1[N+k] \leftarrow 0, \quad k = 1, \ldots, P-1$ $w_1[k] \leftarrow w_1[k-1], \quad k = N, \ldots, 1$ $w_1[0] \leftarrow x[n]$ $q_1[k] \leftarrow q_2[k-1], \quad k = P, \ldots, 1$ $q_1[0] \leftarrow -\sum_{k=1}^{P-1} \frac{a_k^* q_1[P-k]}{a_P^*} - \frac{q_1[P]}{a_P^*} + \sum_{m=0}^{P-1} \frac{b_m'^* w_1[P-1-m]}{a_P^*}$	$w_1[k] \leftarrow w_1[k-1], \quad k = N+P, \ldots, 1$ $w_1[0] \leftarrow x[n]$ $q_1[k] \leftarrow q_1[k-1], \quad k = P, \ldots, 1$ $q_1[0] \leftarrow -\sum_{k=1}^{P-1} \frac{a_k^* q_1[P-k]}{a_P^*} - \frac{q_1[P]}{a_P^*} + \sum_{m=0}^{P-1} \frac{b_m'^* w_1[P-1-m]}{a_P^*} - \sum_{\ell=0}^{P} \frac{b_\ell^* w_1[N+P-\ell]}{a_P^*}$ $y[n] \leftarrow q_1[0]$
Auxiliary TIIR Filter	
$w_2[k] \leftarrow 0, \quad k = 1, \ldots, P-1$ $w_2[0] \leftarrow x[n]$ $q_2[k] \leftarrow 0, \quad k = 1, \ldots, P-1$ $q_2[0] \leftarrow \frac{b_{P-1}'^* w_2[0]}{a_P^*}$	$w_2[k] \leftarrow w_2[k-1], \quad k = P, \ldots, 1$ $w_2[0] \leftarrow x[n]$ $q_2[k] \leftarrow q_2[k-1], \quad k = P, \ldots, 1$ $q_2[0] \leftarrow -\sum_{k=1}^{P-1} \frac{a_k^* q_2[P-k]}{a_P^*} - \frac{q_2[P]}{a_P^*} + \sum_{m=0}^{P-1} \frac{b_m'^* w_2[P-1-m]}{a_P^*}$

terms and thus reduce the total number of multiplies and adds to about $4P$ per input sample. The shifts of the state variables and input delay lines may be simulated by using pointer arithmetic and need not actually be performed.

V. FAST LINEAR-PHASE FIR FILTERS

The implementation of filters as TIIR systems is rather pointless unless there is an advantage that is unavailable to untruncated IIR filters. That advantage is that it is possible to implement exactly linear-phase filters. We present two basic strategies for designing filters based on TIIR filters. The first uses the factorization given in (35), and the second uses the factorization used in (37).

A. Additive Factorization Design Method

We saw in Section II that an FIR filter has linear phase if (11) holds. It is possible to attain such a relation using TIIR filters. Let $H_{\text{FIR}}^{+}(z)$ be a TIIR transfer function such that

$$H_{\text{FIR}}^{+}(z) = \sum_{k=0}^{N} h_k^{+} z^{-k} \tag{41}$$

$$= \frac{B^{+}(z) - z^{-N} B'^{+}(z)}{A^{+}(z)}. \tag{42}$$

We form the time-reversed truncated transfer function

$$H_{\text{FIR}}^{-}(z) = \sum_{k=0}^{N} h_k^{-} z^{-k} \tag{43}$$

$$= \sum_{k=0}^{N} h_k^{+*} z^{k-N} \tag{44}$$

$$= z^{-N} \left\{ H_{\text{FIR}}^{+}\left(\frac{1}{z^*}\right) \right\}^* \tag{45}$$

$$= \frac{z^{-N} \left\{ B^{+}\left(\frac{1}{z^*}\right) \right\}^* - \left\{ B'^{+}\left(\frac{1}{z^*}\right) \right\}^*}{\left\{ A^{+}\left(\frac{1}{z^*}\right) \right\}^*} \tag{46}$$

$$= \frac{-z B'^{-}(z) + z^{-N} B^{-}(z)}{A^{-}(z)} \tag{47}$$

where the "+" and "−" superscripts denote "forward" and "reverse-conjugate" filter, respectively, and the "*" superscript denotes complex conjugation, as usual. Thus, comparing with (8) and (9), we have

$$A^{-}(z) = 1 + a_1^* z + \cdots + a_P^* z^P \tag{48}$$

$$B^{-}(z) = b_0^* + b_1^* z + \cdots + b_P^* z^P \tag{49}$$

and

$$B'^{-}(z) = b_0'^* + b_1'^* z + \cdots + b_{P-1}'^* z^{P-1}. \tag{50}$$

We have assumed that $B^{+}(z)$ and $B'^{-}(z)$ have degrees P and $P-1$, respectively. If we assume that $H_{\text{FIR}}^{+}(z)$ is a stable TIIR filter, then $H_{\text{FIR}}^{-}(z)$ is an unstable TIIR filter whose hidden modes are conjugate reciprocals of those of $H_{\text{FIR}}^{+}(z)$. An example implementation for $H_{\text{FIR}}^{-}(z)$ is given in Table II. There, we have normalized the numerator and denominator of the filter by dividing through by a_P^*.

Using the arbitrary time shift $M \geq 0$ and phase shift ψ, we define the filter

$$H_{\mathrm{LPFIR}}(z) \triangleq H^{+}_{\mathrm{FIR}}(z) + e^{j\psi} z^{-M} H^{-}_{\mathrm{FIR}}(z) \quad (51)$$

$$= \sum_{k=0}^{N} h^{+}_{k} z^{-k} + e^{j\psi} \sum_{k=0}^{N} h^{+*}_{k} z^{k-N-M}. \quad (52)$$

We note that this new FIR filter of length $M + N + 1$ is invariant with respect to reversing the order, conjugating the coefficients, and multiplying by the phase factor $\exp(j\psi)$. Equation (11) holds, and thus, $H_{\mathrm{LPFIR}}(z)$ is a linear-phase FIR filter. We may see the linear-phase property more directly by first noticing that on the unit circle

$$H^{-}_{\mathrm{FIR}}(e^{j\omega}) = e^{-jN\omega} \{H^{+}_{\mathrm{FIR}}(e^{j\omega})\}^{*} \quad (53)$$

so that

$$H_{\mathrm{LPFIR}}(e^{j\omega}) = H^{+}_{\mathrm{FIR}}(e^{j\omega}) + e^{j\psi} e^{-jM\omega} H^{-}_{\mathrm{FIR}}(e^{j\omega}) \quad (54)$$

$$= e^{-j[(N+M)\omega+\psi]/2} \cdot \mathrm{Re}\{e^{j[(N+M)\omega+\psi]/2} H^{+}_{\mathrm{FIR}}(e^{j\omega})\}. \quad (55)$$

Equation (12) gives us the result

$$\tau_d = \frac{M+N}{2}. \quad (56)$$

It is rather difficult to design fast FIR filters to a given set of specifications using this additive factorization design method because it is not intuitively obvious how to control the magnitude $|H_{\mathrm{LPFIR}}(e^{j\omega})|$ due to (55). Nonetheless, for certain impulse response waveforms that are well characterized, this technique provides a useful tool for designing corresponding fast FIR filters to realize them. Some examples are given in Section VI-B.

B. Magnitude-Squared Design Method

The following method has perhaps the greater utility for designing fast linear-phase FIR filters in digital signal processing applications. We begin by choosing a desired nonnegative real-valued transfer function $H^2(z) > 0$ for which there exists a stable transfer function $H(z)$ such that

$$H^2(e^{j\omega}) \triangleq |H(e^{j\omega})|^2 \quad (57)$$

$$= H(e^{j\omega})^{*} H(e^{j\omega}) \quad (58)$$

and polynomials $A^{+}(z)$ and $B^{+}(z)$ such that

$$H(z) = \sum_{k=0}^{\infty} h_k z^{-k} \quad (59)$$

$$= \frac{B^{+}(z)}{A^{+}(z)}. \quad (60)$$

If the coefficients of $A^{+}(z)$ and $B^{+}(z)$ are real, then (57) yields

$$H^2(e^{j\omega}) = H(e^{j\omega}) H(e^{-j\omega}). \quad (61)$$

We form the TIIR transfer function

$$H^{+}_{\mathrm{FIR}}(z) = \sum_{k=0}^{N} h_k z^{-k} \quad (62)$$

$$= \frac{B^{+}(z) - z^{-N} B'^{+}(z)}{A^{+}(z)} \quad (63)$$

as in (42). Similarly, we form $H^{-}_{\mathrm{FIR}}(z)$, as in (47). We see immediately, in light of (53), that the filter

$$H^2_{\mathrm{FIR}}(z) \triangleq H^{+}_{\mathrm{FIR}}(z) H^{-}_{\mathrm{FIR}}(z) \quad (64)$$

has the property that

$$H^2_{\mathrm{FIR}}(e^{j\omega}) = e^{-jN\omega} |H^{+}_{\mathrm{FIR}}(e^{j\omega})|^2 \quad (65)$$

and this is obviously a linear-phase filter with group delay

$$\tau_d = N. \quad (66)$$

This filter may be simply implemented as a cascade of $H^{+}_{\mathrm{FIR}}(z)$ and $H^{-}_{\mathrm{FIR}}(z)$, which may be implemented by (25) and Table II, respectively.

The relationship between $H^2_{\mathrm{FIR}}(z)$ and $H^2(z)$ is seen by considering the cyclic convolution [6]

$$H^{+}_{\mathrm{FIR}}(e^{j\omega}) = \frac{1}{2\pi} \int_{0}^{2\pi} H(e^{j\theta}) W_N[e^{j(\omega-\theta)}) \, d\theta \quad (67)$$

where

$$W_N(e^{j\omega}) = \frac{\sin\{(N+\frac{1}{2})\omega\}}{\sin\left(\frac{\omega}{2}\right)}. \quad (68)$$

The filter length N for $H^{+}_{\mathrm{FIR}}(z)$ and $H^{-}_{\mathrm{FIR}}(z)$ must be chosen long enough so that the blurring induced by the periodic convolution of $H(e^{j\omega})$ by $W_N(e^{j\omega})$ does not induce too much distortion in the frequency response. Indeed, disregarding quantization noise for the moment, as N grows to infinity, $W_N(e^{j\omega})$ approaches an impulse centered at $\omega = 0$, and $H^{+}_{\mathrm{FIR}}(z)$ converges to $H(z)$. There are constraints due to (40) on the filter length, which are considered in the next section.

We notice that the phase of the filter $H(z)$ utilized in the design of $H^2_{\mathrm{FIR}}(z)$ is irrelevant, and thus, IIR filters, which are often considered to have excessive phase distortion near sharp cutoffs, may be used. Thus, any stable discrete-time IIR filter design such as elliptic, Chebyshev, and Butterworth filters may be chosen as $H(z)$ in (57) and transformed into a magnitude-squared filter with linear phase. Additionally, the fact that the magnitude response for $H^2(z)$ is twice the distance from 0 dB compared with the magnitude response of $H(z)$ implies that constraints in the stopband of $H(z)$ need only be half the desired decibel design specification, whereas on the other hand, the ripples in the passband of $H(z)$ must be better than half the desired decibel design specification.

C. A Refined Truncation Algorithm

In both the additive and multiplicative designs of the previous two sections, the length N is constrained by the growth of quantization error in the unstable, reverse-conjugate filter $H^-_{\text{FIR}}(z)$. This bound is given in (40). We note that it is the most *stable* hidden mode of $H^+_{\text{FIR}}(z)$, which gives rise to the most unstable hidden mode of $H^-_{\text{FIR}}(z)$ due to the conjugate-reciprocal correspondence between their modes. These problematic conjugate-reciprocal modes may restrict the direct implementation in Table II to undesirably short lengths. One way around this problem is to set a significance floor λ_S above the floor for numerical precision λ_P so that errors with magnitude less than λ_S are insignificant, and quantization errors are bounded by λ_P. Let $H(z)$ be the untruncated IIR impulse response associated with $H^+_{\text{FIR}}(z)$ as in (60). Since $H(z)$ is stable, we have $|p_k| < 1$, where the p_k's are the poles of $H(z)$ and, consequently, the hidden modes of $H^+_{\text{FIR}}(z)$. The hidden modes of $H^-_{\text{FIR}}(z)$ are thus $1/p_k^*$ and are unstable. We perform a partial fraction expansion on $H(z)$ as in (28) and (29). We then observe the impulse response for each partial fraction. We set the cut-off point N_k for the kth partial fraction response to be the smallest time after which the maximum impulse response becomes insignificant, i.e., $\forall n > N_k$, $|\mu h_{k,n}| \leq \lambda_S$, where μ is the largest magnitude input. We may solve

$$|h_{k,N_k}| = \frac{\lambda_S}{\mu} \tag{69}$$

numerically for N_k by using (29) or by using the approximation

$$h_{k,n} \approx \sum_{\ell=0}^{M_k} \frac{b_{k,\ell}(n-\ell)^{M_k-1}p_k^{n-\ell}}{(M_k-1)!} \tag{70}$$

$$\approx n^{M_k-1}p_k^n \sum_{\ell=0}^{M_k} \frac{b_{k,\ell}p_k^{-\ell}}{(M_k-1)!} \tag{71}$$

$$= B_k n^{M_k-1}p_k^n \tag{72}$$

for large n, where B_k is implicitly defined by (71) and (72). For $M_k = 1$, we have exactly

$$N_k = \frac{\log\left(\left|\frac{\lambda_S}{\mu B_k}\right|\right)}{\log(|p_k|)}. \tag{73}$$

Thus, if $N_k < N$, we may truncate the kth partial fraction response at N_k instead of N without losing significance. However, since $H(z)$ is stable and responses due to the kth partial fraction beyond the N_kth time step are below the significance floor, this does not make any difference. The refinement comes from implementing $H^-_{\text{FIR}}(z)$ as a sum of reversed partial fractions based on (28) and (47) but with the kth partial fraction truncated after $n = N_k$ samples instead of $n = N$ if $N_k < N$. Thus, the unstable mode of $H^-_{\text{FIR}}(z)$ due to the pole at $1/p_k^*$ only needs to grow from having an initial magnitude above the significance floor λ_S and has less time to accumulate exponentially growing quantization noise. We have

$$H^-_{\text{FIR}}(z) = \sum_{k=1}^{N_p} \frac{-zB_k'^-(z) + z^{-N_k'}B_k^-(z,N_k')}{(1-p_k^*z)^{M_k}} \tag{74}$$

where

$$N_k' = \begin{cases} N_k, & N_k \leq N \\ N, & N_k > N. \end{cases} \tag{75}$$

Equation (39) indicates how much quantization noise will accumulate due to the truncated response of length $N = N_k$ of the kth partial fraction of $H^-_{\text{FIR}}(z)$. Assuming that quantization error occurs on the order of the precision floor λ_P, we have $\sigma_\nu^2 \approx \lambda_P^2$. In addition, the significance floor sets a convenient level of noise tolerance; therefore, that may set $\sigma_{\text{tol}}^2 = \lambda_S^2$. We may recast (39) as

$$\lambda_S^2 \geq \lambda_P^2 \frac{1-|p_k|^{-4N_k'}}{1-|p_k|^{-2}} \tag{76}$$

$$\approx \lambda_P^2 \frac{|p_k|^{-4N_k'}}{|p_k|^{-2}-1} \tag{77}$$

and thus

$$\lambda_P \leq \lambda_S |p_k|^{2N_k'-1}\sqrt{1-|p_k|^2}. \tag{78}$$

Combining (73) and (78), assuming $M_k = 1$, we arrive at

$$\lambda_P \leq \frac{\lambda_S^3\sqrt{1-|p_k|^2}}{\mu^2|p_k|\,|B_k|^2}. \tag{79}$$

We see that the precision, in bits, for the state variables must be about three times greater than the precision of the data in order to prevent significant noise accumulation. This result is intuitive since the significant part of the impulse response must span the dynamic range specified by the largest magnitude output (which may be assumed to be normalized to 1) and λ_S over a period of N_k' samples, whereas the noise in $H^-_{\text{FIR}}(z)$ may grow over a period of up to $2N_k'$ samples using the same dynamics. The preceding analysis is not generally applicable when dealing with floating-point quantities because the range of significance varies with the exponent. Assuming a constant-energy input, we see that the analysis still holds since the significance and precision floors are approximately constant at steady state.

1) Unit-Magnitude Mode TIIR Filters: A third class of linear-phase filters consisting of pure TIIR responses is possible. If we choose $H_u(z)$ to be a TIIR filter of length N such that all of its (hidden) modes are on the unit circle and each mode has multiplicity one, then we are assured that $H_u(z)$ will have linear phase. This may be seen by observing that such a filter must satisfy (11). If there are modes with multiplicity greater than one, then (11) is not generally satisfied. However, care may be taken to place the zeros so that (11) holds.

Applying the analysis of the previous section, we see from (73) that the N_k's for this type filter will all be infinity. In this case, it is not necessary to reset the state variables very often because the growth of quantization error will be additive and not exponential.

From a worst-case point of view, the number of samples in which error in the least significant bit may accumulate before reaching the significance floor λ_s is approximately

$$N = \frac{\lambda_s}{\lambda_p} \tag{80}$$

assuming that there is 1 b worth of error in the LSB per sample processed. If the noise is zero-mean, then the accumulated error is a random walk whose standard error grows proportionally to the square root of the number of samples processed. We would like the standard error to be some distance away from the least significant bit of the output signal. For an error accumulation such that the three-sigma error is below the significance floor, we have the relation

$$3\lambda_p\sqrt{N} < \lambda_s \tag{81}$$

giving

$$N < \left(\frac{\lambda_s}{3\lambda_p}\right)^2 \tag{82}$$

$$\approx \frac{4^G}{10} \tag{83}$$

where G is the number of guard bits. Hence, we see that TIIR filters with hidden modes only on the unit circle have desirable stability properties over the filters described in the previous two sections.

Since unit-magnitude mode TIIR filters are quasistable and do not need to be reset every N cycles, the number of computations necessary to stabilize them may be reduced by performing the state-variable reset infrequently. Since an FIR filter has finite memory, it is sufficient to run two parallel filters starting only N steps before the state-variable transfer.

The impulse responses of TIIR filters with unit-magnitude hidden modes are sums of modal responses of the form

$$h_k[n] = P_k(n)p_k^n \tag{84}$$

$$= P_k(n)\,\exp(j\omega_k n) \tag{85}$$

for $n = 0, \cdots, N$, where p_k is the kth unique pole, and $P_k(n)$ is an $m_k - 1$th degree polynomial, where m_k is the multiplicity of the kth pole; these are truncated polynomials times complex exponentials.

Some examples of TIIR filters of the unit-magnitude-modes class are given in Section VI-B.

D. Computational Cost

Assuming that $H^+_{\mathrm{FIR}}(z)$ is a stable filter, we may implement it with $3P + 1$ multiplies and $3P$ adds per sample. $H^-_{\mathrm{FIR}}(z)$ is then unstable and must be calculated with parallel state variables as outlined in Section IV, Table II, and the previous section. Table II reflects a few changes from the algorithm outlined in Table I that are necessary when implementing the reversed filter $H^+_{\mathrm{FIR}}(z)$. The cost for $H^-_{\mathrm{FIR}}(z)$ is about $4P$ adds and multiplies per sample, as discussed in Section IV. We see, then, that the number of operations is about $7P$ multiplies and adds per sample.

It is interesting to note that implementing a partial fraction (parallel) structure for a transfer function as opposed to a direct implementation, as in (25), does not significantly alter the number of multiply accumulates used, as long as the characteristic polynomials are monic and the numerators have degree one less than the denominator. However, each separate factor requires an additional input delay line of length $N_k + M_k$, which might be avoided by clever indexing. In addiiton, the algorithmic overhead may increase, not to mention a greater amount of design effort. The advantage is that greater numerical stability is achieved and that N may then be chosen arbitrarily large. Additionally, each mode is present only where it is above the significance threshold λ_S. Of course, a full factorization is not necessary; it is sufficient to group together modes with the same magnitude or to implement only the troublesome modes separately.

The unit-magnitude-mode, linear-phase filters described in Section V-C1 are particularly computationally efficient. Since such filters are intrinsically self-Hermitian, there is no need to implement a mirror filter as in the additive and magnitude-squared design methods. Additionally, the resetting only needs to be implemented infrequently and may be absorbed into the overhead cost. Hence, the computational cost should be approximately $3P$ operations per sample plus the refresh overhead.

VI. Simulations and Examples

To illustrate the basic idea of a TIIR system, we examine the system

$$H(z) = \frac{z^2}{z^2 - 1.9z + 0.98} = \frac{B^+(z)}{A^+(z)} \tag{86}$$

we wish to truncate after $N = 300$ samples; thus, we wish to have an FIR response of 301 steps. We first form the tail-canceling polynomial as in (24). We perform synthetic division on $z^{300}B(z)$ by $A(z)$ and obtain the remainder

$$B'^+(z) = -0.162126z + 0.139770 \tag{87}$$

so that $H^+_{\mathrm{FIR}}(z)$ of (42) is defined. We plot the impulse response of the direct implementation given in (25) in Fig. 1(a) and (b). We see that at step $n = 301$, the tail of the response has been subtracted off and the response magnitude drops by about 115 dB. Due to quantization errors, there is a residual response. Fig. 1(c) and (d) show the same system but implemented using the algorithm presented in Section IV and Table I. Here, the system is forced to have a finite memory, and thus, the residual response is completely canceled at time step $n = 600$ because the state variables are refreshed every $N = 300$ time steps.

We next form the reverse-conjugate system $H^-_{\mathrm{FIR}}(z)$ as described by (47) and arrive at

$$H^-_{\mathrm{FIR}}(z) = \frac{-zB'^-(z) + z^{-N}B^-(z)}{A^-(z)} \tag{88}$$

$$= \frac{-0.139\,770z^2 + 0.162\,126z - z^{-N}}{0.98z^2 - 1.9z + 1} \tag{89}$$

$$= \frac{-0.142\,622z^2 + 0.165\,435z - 1.020\,408z^{-N}}{z^2 - 1.938\,776z + 1.020\,408} \tag{90}$$

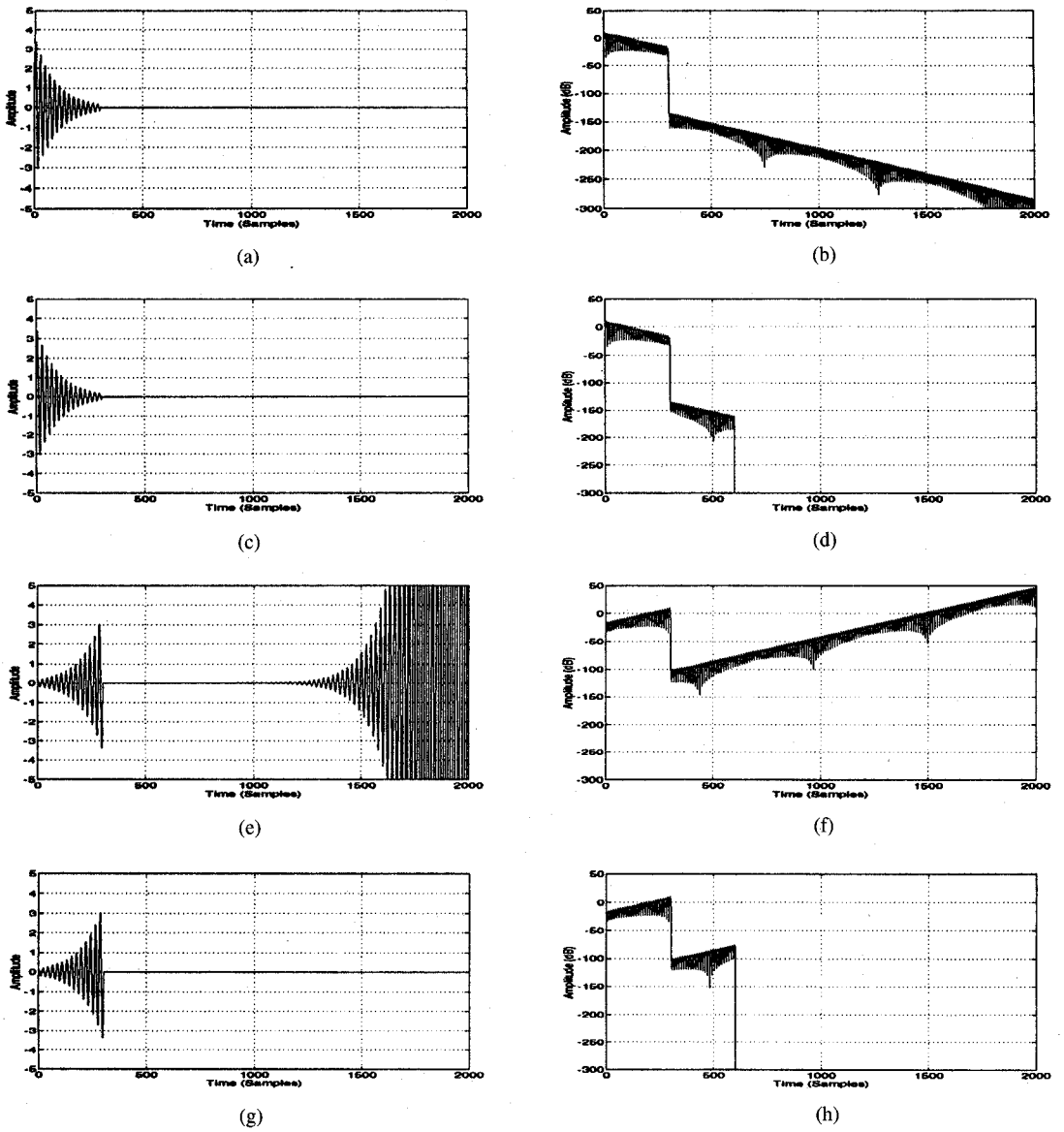


Fig. 1. TIIR impulse response examples.

where, in the last equation, we have normalized by 0.98 in order to have a monic characteristic polynomial. This system has unstable hidden modes, and its impulse response is plotted in Fig. 1(e) and (f). The tail is canceled with about 125-dB attenuation, but the quantization noise grows without bound to overwhelm the signal eventually. Fig. 1(g) and (h) show how the state variable refresh technique completely cancels the quantization noise before it becomes significant.

TABLE III
ELLIPTIC FILTER COEFFICIENTS FOR THE EXAMPLE IN SECTION VI-A

Elliptic Filter Coefficients			
a_0	1.00000000	b_0	0.05149489
a_1	−5.20086294	b_1	−0.25706694
a_2	11.46455205	b_2	0.57645267
a_3	−13.68525876	b_3	−0.74102189
a_4	9.32002688	b_4	0.57645267
a_5	−3.43103178	b_5	−0.25706694
a_6	0.53331689	b_6	0.05149489

A. Elliptic-Filter-Based FFIR Design Example

We illustrate here the design of a lowpass linear-phase FFIR filter using the magnitude-squared technique outlined in Section V-B. We wish to design a filter that meets the following criteria:

1) passband (0.00, 0.10) in fractions of $f_s/2$ with at most 0.080 dB maximum peak-to-peak ripple, where f_s is the sampling frequency,
2) stopband (0.11, 1.00) with at least 50 dB attenuation,
3) linear phase,
4) 15 b of fixed-point significance in the mantissa,
5) maximum input amplitude of $\mu = 1.0$.

Because of the sharp transition band in the interval (0.10, 0.11), we select an elliptic filter as the basis for our design [7]. We realize that our choice of $H(z)$ only needs to meet half the specifications of 1) and 2) above, and that 3) is irrelevant. Using Matlab, we find that an order 6 filter suffices to give

TABLE IV
ELLIPTIC FILTER POLES, MAGNITUDES, AND DECAY TIMES TO SIGNIFICANCE LEVEL λ_S FOR THE EXAMPLE IN SECTION VI-A

k	Poles	Residue	N_k	$\lambda_P(dB)$
1	$0.93560805 + j0.31706231$	$0.00710587 + j0.01100914$	497	-211.6943
2	$0.93560805 - j0.31706231$	$0.00710587 - j0.01100914$	497	-211.6943
3	$0.88941945 + j0.28954430$	$-0.06609826 + j0.02150355$	116	-233.0556
4	$0.88941945 - j0.28954430$	$-0.06609826 - j0.02150356$	116	-233.0556
5	$0.77540397 + j0.15290736$	$0.06436785 - j0.21546997$	38	-247.2166
6	$0.77540397 - j0.15290736$	$0.06436785 + j0.21546997$	38	-247.2166
Direct term: $h_0 = 0.05149489$				

us a filter $H(z)$ with the desired properties such that $H^2(z)$ satisfies the above criteria. The arguments to Matlab's *ellip* function are as follows:

- $N = 6$,
- passband ripple $= 0.035$ dB,
- stopband attenuation $= 25$ dB,
- cut-off frequency $= 0.10$ (normalized units).

We have allowed an additional 0.005-dB margin in the passband to anticipate Gibbs ringing due to sequence truncation expressed in (67). For the form given in (5), the coefficients are given in Table III. Since the multiplicity of each pole is one, we may use (73) to calculate the decay times N_k to the significance floor. We round up in each case. The poles are given in Table IV as well as the magnitude and N_k for each pole. Because the coefficients come in complex-conjugate pairs, we may rearrange $H(z)$ into three second-order real terms so that

$$H(z) = h_0 + \frac{b_{1,1}z^{-1} + b_{1,2}z^{-2}}{1 + a_{1,1}z^{-1} + a_{1,2}z^{-2}} + \frac{b_{2,1}z^{-1} + b_{2,2}z^{-2}}{1 + a_{2,1}z^{-1} + a_{2,2}z^{-2}} + \frac{b_{3,1}z^{-1} + b_{3,2}z^{-2}}{1 + a_{3,1}z^{-1} + a_{3,2}z^{-2}} \tag{91}$$

where the coefficients are given in Table V. We then implement each term separately in the truncated form given in (25) to form $H^+_{\mathrm{FIR}}(z)$ using each term's decay-to-significance times N_1, N_3, and N_5 for the cutoffs. The coefficients are calculated by using synthetic division and are listed in Table V. We choose $N = 497$ as the cutoff point for truncating the filter since this is the largest N_k. Thus, we may implement the forward filter in (92), shown at the bottom of the page. Using (47), we form (93), shown at the bottom of the page, from the corresponding parts in (92). Notice that each term is delayed by an appropriate $N - N_k$ steps in order to align the phases of the response properly. To implement this filter, we use the algorithm of Table II and Section IV due to the unstable hidden modes of each dynamic term. We may wish to normalize the coefficients by $a_{k,2}$ so that the characteristic polynomials are monic in each term. At this point, we form $H^2_{\mathrm{FIR}}(z)$ by cascading the output of $H^-_{\mathrm{FIR}}(z)$ into the input of $H^+_{\mathrm{FIR}}(z)$. This order may be preferable since the state-variable reinitialization scheme used in Section IV may introduce some discontinuities near or below the significance floor λ_S. Thus, postfiltering by $H^+_{\mathrm{FIR}}(z)$ attenuates any such additional noise, however insignificant.

We show in Fig. 2 the results of our filter implementation. Fig. 2(a)–(c) show the ideal untruncated $H^2(z)$ response from our design. Its decibel response is simply double that of $H(z)$ from (91). Fig. 2(d)–(f) show the truncated response of $H^2_{\mathrm{FIR}}(z) = H^-_{\mathrm{FIR}}(z)H^+_{\mathrm{FIR}}(z)$. Notice the presence of Gibbs overshoot due to the truncation of the impulse response. Fig. 2(g)–(i) show the implementation $H^-_{\mathrm{FIR}}(z)H(z)$ in which

$$H^+_{\mathrm{FIR}}(z) = h_0 + \frac{b_{1,1}z^{-1} + b_{1,2}z^{-2} - b'_{1,1}z^{-(N_1+1)} - b'_{1,2}z^{-(N_1+2)}}{1 + a_{1,1}z^{-1} + a_{1,2}z^{-2}} + \frac{b_{2,1}z^{-1} + b_{2,2}z^{-2} - b'_{2,1}z^{-(N_3+1)} - b'_{2,2}z^{-(N_3+2)}}{1 + a_{2,1}z^{-1} + a_{2,2}z^{-2}} + \frac{b_{3,1}z^{-1} + b_{3,2}z^{-2} - b'_{3,1}z^{-(N_5+1)} - b'_{3,2}z^{-(N_5+2)}}{1 + a_{3,1}z^{-1} + a_{3,2}z^{-2}}. \tag{92}$$

$$H^-_{\mathrm{FIR}}(z) = h_0 + z^{-N-N_1}\frac{-b'_{1,2} - b'_{1,1}z^{-1} + b_{1,2}z^{-N_1} + b_{1,1}z^{-(N_1+1)}}{a_{1,2} + a_{1,1}z^{-1} + z^{-2}} + z^{-N-N_3}\frac{-b'_{2,2} - b'_{2,1}z^{-1} + b_{2,2}z^{-N_3} + b_{2,1}z^{-(N_3+1)}}{a_{2,2} + a_{2,1}z^{-1} + z^{-2}} + z^{-N-N_5}\frac{-b'_{3,2} - b'_{3,1}z^{-1} + b_{3,2}z^{-N_5} + b_{3,1}z^{-(N_5+1)}}{a_{3,2} + a_{3,1}z^{-1} + z^{-2}} \tag{93}$$

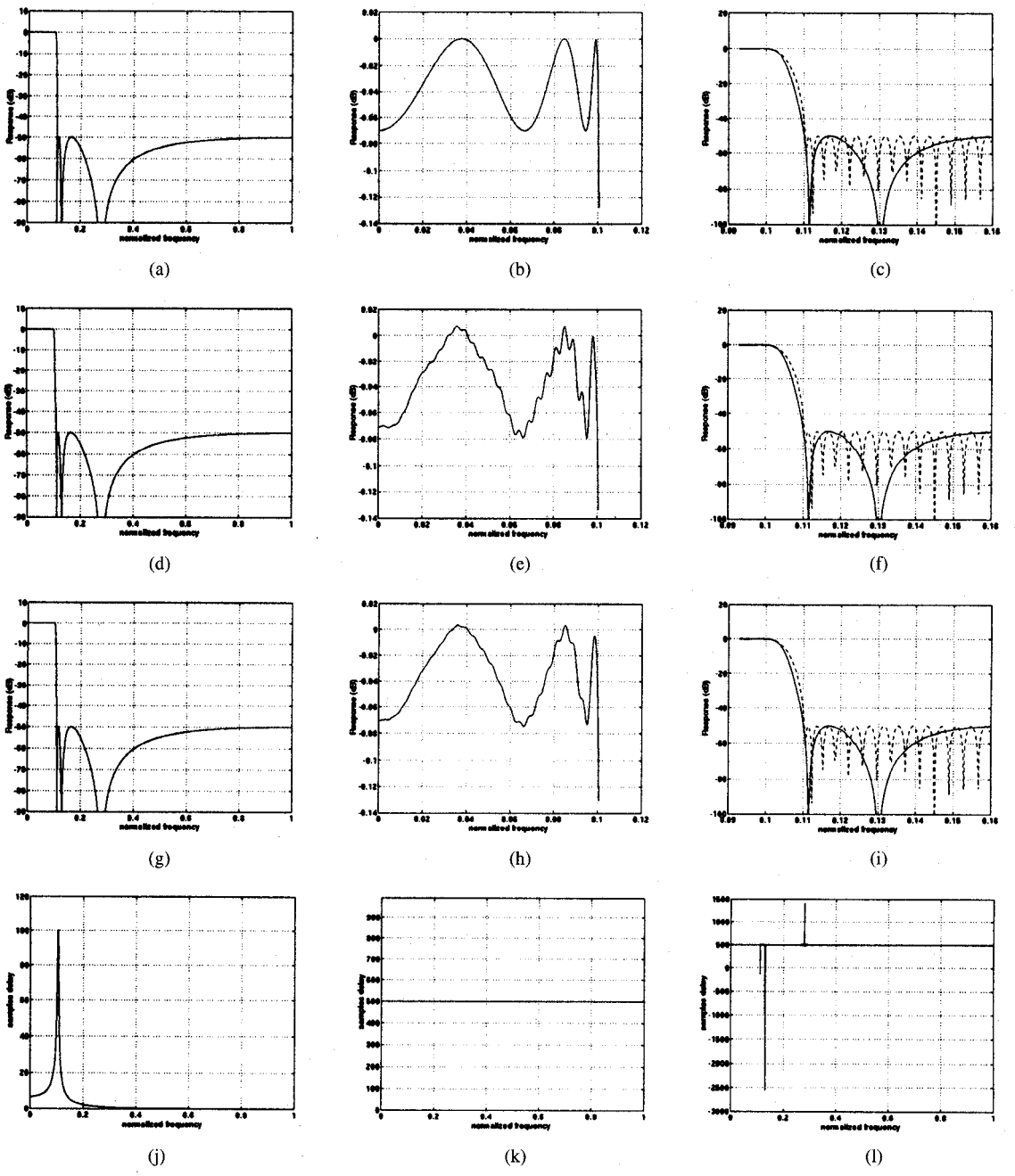


Fig. 2. Filter responses from the linear-phase elliptic filter design example in Section VI-A.

TABLE V
COEFFICIENTS FOR THE MODAL DECOMPOSITION OF THE ELLIPTIC FILTER IN (91) AND (92)

	$\ell = 1$	$\ell = 2$
$a_{1,\ell}$	−1.87121609	0.97589092
$b_{1,\ell}$	0.01421174	−0.02027779
$b'_{1,\ell}$	6.08847078e−05	−5.74152962e−05
$a_{2,\ell}$	−1.77883890	0.87490286
$b_{2,\ell}$	−0.13219652	0.10512570
$b'_{2,\ell}$	−3.88583695e−06	−1.38234153e−05
$a_{3,\ell}$	−1.55080795	0.62463198
$b_{3,\ell}$	0.12873570	−0.03392829
$b'_{3,\ell}$	5.80618310e−05	−4.35418795e−05

the tail of the forward response is not truncated. The response is smoother, but there are some slight deviations from phase linearity due to the asymmetry. The error due to the tail response of $H(z)$ should be small, however, since the tail of each mode for $H^{+}_{\mathrm{FIR}}(z)$ is cut off at the significance floor λ_S. The group delays for $H(z)$, $H^{2}_{\mathrm{FIR}}(z)$, and $H^{-}_{\mathrm{FIR}}(z)H(z)$ are shown in Fig. 2(j)–(l).

B. Windows and Polynomials

Unit-magnitude mode TIIR responses are convenient for generating well-known impulse response functions for computing weighted averages of an input sequence. In signal processing applications, an exponential window is often undesirable for averaging highly nonstationary data because of the long tail. Finite-length windows with polynomial terms may be generated using multiple poles at $z = 1$ and using (29). The terms due to the binomial coefficients generate the

$$H_{\text{Hanning}}(z) = \frac{1}{2}\left[\frac{1-z^{-N}}{1-z^{-1}} - \frac{1-\cos\left(\frac{2\pi}{N-1}\right)(z^{-1}+z^{-N})+z^{-(N+1)}}{1-2\cos\left(\frac{2\pi}{N-1}\right)z^{-1}+z^{-2}}\right]. \tag{96}$$

polynomial coefficients. The FFIR algorithm provides a cost-effective method for implementing such weighting sequences. For example, the simple N-point rectangular window may be generated as a quasistable TIIR filter with filter dynamics

$$H_{\text{rectangular}}(z) = \frac{1-z^{-N}}{1-z^{-1}}. \tag{94}$$

The $2N$-point Bartlett window is

$$H_{\text{Bartlett}}(z) = \frac{1}{N}\frac{1-z^{-N}-z^{-(N+1)}+z^{-(2N+1)}}{1-2z^{-1}+z^{-2}}. \tag{95}$$

The N-point Hanning window, which is also known as the *Hann* window, is as in (96), shown at the top of the page. The N-point Hamming window is

$$H_{\text{Hamming}}(z) = 0.54\frac{1-z^{-N}}{1-z^{-1}} - 0.46\frac{1-\cos\left(\frac{2\pi}{N-1}\right)(z^{-1}+z^{-N})+z^{-(N+1)}}{1-2\cos\left(\frac{2\pi}{N-1}\right)z^{-1}+z^{-2}}. \tag{97}$$

An example of a useful polynomial impulse response is the Kay window [2], which is used for statistically efficient frequency estimation based on averaged phase differences with weighting coefficients

$$w_{\text{Kay}}[n] = \frac{6N}{N^2-1}\left\{\frac{n}{N} - \left(\frac{n}{N}\right)^2\right\} \tag{98}$$

for $1 \le n \le N-1$ [see (99), shown at the bottom of the page]. This implementation requires only six adds and two multiplies, independent of N.

VII. Applications

We list here some of the many possible applications for TIIR filters.

A. Digital Speaker Crossover Networks and Mixers for Digital Audio Applications

In the past, commercial digital audio systems designers have had to choose between efficient, nonlinear-phase IIR filters and linear-phase, but computationally expensive, FIR filters. Mixing the outputs from nonlinear-phase filters may result in unwanted nulls in the combined frequency response due to cancellation of signals with different frequency-dependent delays. The linear-phase TIIR algorithm combines the best of these features from both types of filters.

B. Multirate Filtering

One disadvantage of the fast FIR filtering methods is that although they are computationally efficient, they are limited to hop sizes of one. Multirate [5] or STFT techniques, which downsample the data stream to account for a reduction in bandwidth, can be quite efficient, thus reducing the relative advantage of TIIR filtering over FIR filters. However, polyphase interpolating filters for upsampling in multirate systems may be easily implemented as TIIR filters, providing a great savings in computation, thus gaining the best features of both filter design approaches. The development of applications of TIIR theory to multirate filter design is beyond the scope of this paper.

C. High-Resolution, Optimal Frequency Estimator

Kay derived the optimal weighting sequence for estimating the frequency of a pure-frequency sinusoid in Gaussian noise from finite phase differences [8]. The window given in (98) attains the Cramér–Rao bound for moderate SNR. Equation (99) shows how a running average may be computed using only two multiplies and six adds per sample, independent of the window length. Such a system may be used for computing a running estimate of instantaneous frequency.

D. Polynomial Impulse Responses

Sometimes a finite-length window is desired for processing nonstationary signals where long exponential tails are undesirable. Running windows such as the Bartlett, Hamming, or Hanning window give a more localized and symmetric average than a decaying exponential. Additionally, polynomial responses with sinsuoidal modulation may be constructed easily, as seen in Section V-C1.

VIII. Summary

The power of the fast FIR algorithm stems from the ability to form complicated FIR sequences as the quotient of two polynomials with relatively few nonzero coefficients. The resulting cancellation of all the dynamic modes yields the desired FIR sequence. This may be thought of as "filter compression," in which low-entropy redundancies in certain

$$H_{\text{Kay}}(z) = \frac{6}{N(N^2-1)} \times \frac{(N-1)z^{-1}-(N+1)z^{-2}+(N+1)z^{-N-1}-(N-1)z^{-N-2}}{(1-z^{-1})^3}. \tag{99}$$

kinds of FIR sequences are encoded using generating functions based on IIR dynamics. We utilize unstable as well as stable IIR dynamics in creating our TIIR responses by exploiting a state-variable-resetting technique, which takes advantage of the fact that FIR responses have finite memory. Another technique introduced here is the use of synthetic division to generate a tail-canceling complement filter. Finally, the technique described in Section V-B allows the straightforward use of conventional IIR filter design methods for creating linear-phase fast FIR filters.

REFERENCES

[1] T. Saramäki and A. T. Fam, "Properties and structures of linear-phase FIR filters based on switching and resetting of IIR filters," in *Proc. IEEE Int. Symp. Circuits Syst.,* Piscataway, NJ, 1990, pp. 3271–3274.

[2] S. Kay, "Statistically/computationally efficient frequency estimation," in *Proc. Int. Conf. Acoust., Speech, Signal Processing*, Piscataway, NJ, 1988, pp. 2292–2295.

[3] A. T. Fam, "FIR filters that approach IIR filters in their computational efficiency," in *Proc. 21st Asilomar Conf. Signals, Syst., Comput.*, Pacific Grove, CA, 1987, pp. 28–30.

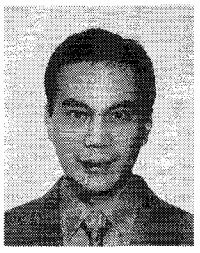
[4] A. Papoulis, *Signal Analysis.* New York: McGraw-Hill, 1977.

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

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

[7] A. Gray and J. Markel, "A computer program for designing elliptic filters," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. ASSP-24, pp. 529–538, Dec. 1976.

[8] S. Kay, "A fast and accurate single frequency estimator," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. 37, pp. 1987–1990, Dec. 1989.



Avery Wang received the B.S. degree in mathematics, with distinction, the M.S. degree in mathematics, and the M.S. degree in electrical engineering, all in 1988 from Stanford University, Stanford, CA. He was with the Center for Computer Research in Music and Acoustics, Stanford University, and received the Ph.D. degree in electrical engineering in 1994.

He is currently with Chromatic Research, Sunnyvale, CA, doing algorithm design and implementation. His research interests include auditory source separation, statistical signal processing, and music synthesis. Other interests include molecular biology, neurosciences, mathematics, and languages.

Dr. Wang won a National Science Foundation Graduate Fellowship Award in 1988 and a Fulbright Graduate Scholarship in 1990, with which he spent two years at the Ruhr-Universität Bochum, Germany, studying neurosciences and computational models of auditory perception.

Julius O. Smith, III (M'82) received the B.S.E.E. degree from Rice University, Houston, TX, in 1975. He received the M.S. and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA, in 1978 and 1983, respectively. His Ph.D. research involved the application of digital signal processing and system identification techniques to the modeling and synthesis of the violin, clarinet, reverberant spaces, and other musical systems.

From 1975 to 1977, he worked in the Signal Processing Department, ESL, Sunnyvale, CA, on systems for digital communications. From 1982 to 1986, he was with the Adaptive Systems Department, Systems Control Technology, Palo Alto, CA, where he worked in the areas of adaptive filtering and spectral estimation. From 1986 to 1991, he was employed at NeXT Computer, Inc., where he was responsible for sound, music, and signal processing software for the NeXT computer workstation. Since then, he has been an Associate Professor at the Center for Computer Research in Music and Acoustics, Stanford University, teaching courses in signal processing and music technology and pursuing research in signal processing techniques applied to music and audio. His home page on the Web can be found at http://www-ccrma.stanford.edu/~jos/.