On Fast FIR Filters Implemented as Tail-Canceling IIR Filters

Avery Wang, Julius O. Smith III

Abstract

We have developed an algorithm based on synthetic division for deriving the transfer function which 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 [1]. 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: one based on the addition and the other on the cascading of anti-phase 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 [2].

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 which 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 N-th order FIR filter requires $N+1$ multiplies and N adds. In certain cases, however, FIR filters may be designed which have an operation count comparable to that of an IIR filter while maintaining the linear phase property.

The set of finite impulse responses which may be efficiently implemented includes those which 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 which 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.

II. Definitions

A general causal N-th order FIR filter consists of a tapped delay line N elements long and a table of $N+1$ impulse response coefficients $\{h_0, \ldots, h_N\}$ 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

$$\begin{aligned} H_{\mathrm{FIR}}(z) &\stackrel{\triangle}{=} h_0 + h_1 z^{-1} + \ldots + h_N z^{-N} && (2)\\ &= z^{-N} C(z), && (3)\end{aligned}$$

where $C(z)$ is the N-th degree polynomial formed by the h_k. On the other hand, a causal P-th 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

$$H_{\text{IIR}}(z) \triangleq \frac{b_0 + b_1 z^{-1} + \ldots + b_P z^{-P}}{1 + a_1 z^{-1} + \ldots + a_P z^{-P}} \quad (5)$$

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

$$\triangleq h_0 + h_1 z^{-1} + h_2 z^{-2} + \ldots, \quad (7)$$

where both

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

and

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

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

Group delay [4] is defined by

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

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., $\arg\left\{H(e^{j\omega})\right\} = K_1\omega + K_2$ 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 non-zero poles cannot have linear phase. However, an FIR filter with coefficients $\{h_0, \ldots, h_N\}$ has linear phase if there exists a ψ such that for all $k \in \{0, \ldots, N\}$

$$h_{N-k} = e^{j\psi} h_k^*, \quad (11)$$

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

$$\delta(f) = \frac{N}{2}. \quad (12)$$

III. Truncated IIR (TIIR) Filters

Consider an FIR filter having a truncated geometric sequence $\{h_0, h_0 p, \ldots, h_0 p^N\}$ 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

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

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

$$H_{\text{FIR}}(z) = h_0 + h_0 p z^{-1} + \ldots + h_0 p^N z^{-N} \quad (14)$$

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

The time-domain recursion for this filter is

$$y[n] = \sum_{k=0}^{N} h_0 p^k x[n-k] \quad (16)$$

$$= p y[n-1] + h_0 \left(x[n] - p^{N+1} x[n-(N+1)] \right). \quad (17)$$

We see that the first formulation, Eqn. (16), requires $N+1$ multiplies and N adds to implement directly, whereas the second formulation, Eqn. (17), requires only 3 multiplies and 2 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 Eqn. (17) still requires a delay line to be maintained, and thus there are no savings in storage. With modern digital signal processing (DSP) chips, however, ring buffers may be implemented with virtually no computational overhead and the full savings in computational cost may be achieved.

Notice that there is a pole-zero cancellation in the representation given by Eqn. (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 Eqn. (15). This method works only for cases in which the multiplicity of each

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

$$\begin{aligned} H'_{\text{IIR}}(z) &= h'_0 z^{-1} + h'_1 z^{-2} + \ldots \\ &\triangleq h_{N+1} z^{-1} + h_{N+2} z^{-2} + \ldots \end{aligned} \tag{18}$$

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 Eqn. (7) by z^N and obtain

$$\begin{aligned} z^N H_{\text{IIR}}(z) &= h_0 z^N + \ldots + h_{N-1} z + h_N \\ &\qquad + h_{N+1} z^{-1} + h_{N+2} z^{-2} + \ldots && (19) \\ &= C(z) + H'_{\text{IIR}}(z) && (20) \\ &= \frac{z^N B(z)}{A(z)} && (21) \\ &= C(z) + \frac{B'(z)}{A(z)}, && (22) \end{aligned}$$

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 $z^N B(z)$ by $A(z)$ and finding the remainder. Thus, $z^N B(z) \equiv B'(z) \pmod{A(z)}$.

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

$$\begin{aligned} H_{\text{FIR}}(z) &= H_{\text{IIR}}(z) - z^{-N} H'_{\text{IIR}}(z) && (23) \\ &= \frac{B(z) - z^{-N} B'(z)}{A(z)}. && (24) \end{aligned}$$

The corresponding system is

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

using the representation in Eqn. (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 P-th 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 $z^{-N} H'_{\text{IIR}}(z)$ from $H_{\text{IIR}}(z)$ is to cancel all the poles in Eqn. (24), which is to be expected since an FIR filter is an all-zero filter.

B. Other Architectures

The direct implementation specified by Eqn. (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

$$H(z) = \sum^{N_p} \sum^{M_k} \frac{C_{k,\ell}}{\ldots} \tag{26}$$

where N_p is the number of distinct poles, and M_k is the multiplicity of the k-th 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 Eqn. (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 n-th order multiplicity term yields a degree $n-1$ polynomial numerator anyway. The impulse response of the k-th 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 $b_{k,\ell}$, $\ell = 0, \ldots, M_k$ 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

$n \equiv 0 \pmod N$	$n \not\equiv 0 \pmod N$
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	
$w_2[k] \leftarrow 0, \quad k = 1, \ldots, P-1$ $w_2[0] \leftarrow x[n]$ $q_2[k] \leftarrow 0, \quad k = 1, \ldots, P-1$ $q_2[0] \leftarrow b_0 w_2[0]$	$w_2[k] \leftarrow w_2[k-1], \quad k = P, \ldots, 1$ $w_2[0] \leftarrow x[n]$ $q_2[k] \leftarrow q_2[k-1], \quad k = P, \ldots, 1$ $q_2[0] \leftarrow -\sum_{k=1}^{P} a_k q_2[k] + \sum_{\ell=0}^{P} b_\ell w_2[\ell]$

TABLE I

Fast FIR algorithm for an unstable Truncated IIR (TIIR) filter.

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 Eqn. (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 $|p_k| > 1$.

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

TABLE II

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

noise signal $\nu[n]$ with zero mean and variance σ_ν^2 input to the system of Eqn. (17).[1] The output equation is then

$$\begin{aligned} y[n] &= py[n-1] + h_0 \left(x[n] - p^{N+1} x[n-(N+1)] \right) + \nu[n] && (30) \\ &= h_0 \sum_{k=0}^{N-1} p^k x[n-k] + \sum_{k=0}^{n} p^k \nu[n-k]. && (31) \end{aligned}$$

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

$$\begin{aligned} \sigma_{\mathrm{acc}}^2[n] &= \sum_{k=0}^{n} |p|^{2k} \sigma_\nu^2 && (32) \\ &= \sigma_\nu^2 \frac{1 - |p|^{2(n+1)}}{1 - |p|^2} && (33) \end{aligned}$$

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 dominates the behavior of the system. A direct implementation of an FIR system, such as in equation (16) does not have noise accumulation problems because of its finite memory. We trade computational cost for noise sensitivity in TIIR systems.

[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

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.

WHAT WILL YOU BUILD? | **sales@docketalarm.com** | **1-866-77-FASTCASE**

fastcase® Smarter legal research.