

Analysis of Root Displacement Interpolation Method for Tunable Allpass Fractional-Delay Filters

Hüseyin Hacıhabiboğlu, *Member, IEEE*, Banu Günel, *Member, IEEE* and Ahmet M. Kondoç, *Member, IEEE*

Abstract—One of the simplest ways of designing allpass fractional-delay filters with maximally-flat group delays is by using the Thiran approximation by which the filter coefficients are calculated using a closed-form equation. However, due to the number of multiplications and divisions involved, the calculation of these coefficients is a computationally costly task and is not suitable for real-time applications. The analysis of a root-displacement based interpolation method used in allpass tunable fractional delays is presented in this article. The method allows continuous adjustments of the approximated fractional delay without the explicit calculation of a new set of filter coefficients. Transient error observed at the output due to the change of filter coefficients is analyzed. The direct and cascade implementations are compared with respect to their transient errors. An example application of the proposed method from the field of model-based sound synthesis is given.

Index Terms—Fractional delay filters, allpass systems, filter interpolation, tunable filters, audio signal processing, transients.

I. INTRODUCTION

FRactional-DELAY filter is the generic name given to the discrete-time filters that simulate non-integer delays [1]. They are used in a diverse set of signal processing applications such as spatial audio synthesis [2], physical modelling [3], speech synthesis [4], synchronisation in wireless telecommunications [5], sample-rate conversion [6], and microphone array processing [7].

There are different varieties of the fractional-delay filters. Most frequently used fractional-delay filters are FIR filters based on Lagrange interpolation [8], [9]. The Farrow structure [5] allows continuously varying the fractional delay using a single parameter. However, the major issue with FIR fractional-delay filters is that, both the magnitude and the phase responses will deviate from the desired response, and these errors can only be reduced at the cost of increasing the filter order. Although using an FIR fractional-delay filter is sufficient for many applications, a group of other applications, among which audio processing is a notable example, require that the original magnitude spectra and the bandwidth of the

input signal remains unaffected. This is possible by using allpass fractional-delay filters. There exist design methods based on optimization for obtaining allpass fractional delay filters [10] which are computationally costly for real-time operation. One of the simplest design methods to obtain maximally-flat delay allpass fractional-delay filters is by using the Thiran approximation [11], [12]. Consider an N^{th} -order allpass IIR filter having the following transfer function:

$$H(z) = \frac{B(z)}{A(z)} = \sum_{k=0}^N a_{(N-k)} z^{-k} \bigg/ \sum_{k=0}^N a_k z^{-k}, \quad (1)$$

where $a_0 = 1$, $B(z)$ and $A(z)$ are the numerator and denominator polynomials, respectively. The phase response of this IIR filter can be expressed as:

$$\angle H(\omega) = -N\omega + 2 \arctan \left(\frac{\sum_{k=0}^N a_k \sin k\omega}{\sum_{k=0}^N a_k \cos k\omega} \right), \quad (2)$$

where k is the coefficient index. For an N^{th} -order allpass filter to have a maximally-flat group delay at d.c. equal to an arbitrary delay, D , the approximation error,

$$\epsilon(\omega) = -\tan(\omega D) - \tan(\angle H(\omega)), \quad (3)$$

must vanish at $\omega = 0$ together with N of its derivatives. In other words, the slope of the phase response must be equal to the desired arbitrary non-integer delay at d.c.

Using the Thiran approximation, the coefficients of an N^{th} -order allpass fractional-delay filter that approximates the total fractional delay, $D > N - 1$, can be calculated using the following closed-form formula [1]:

$$a_k = (-1)^k \binom{N}{k} \prod_{n=0}^N \frac{D - N + n}{D - N + n + k}. \quad (4)$$

The filter with the given coefficients is guaranteed to be stable, and to have maximally-flat group delay corresponding to the desired fractional delay, D , at low frequencies [11]. Using (4), the calculation of the k^{th} coefficient requires $(2N + 1)$ multiplications and a single division excluding the computation of the binomial term. For an N^{th} -order Thiran filter this amounts to $(2N^2 + N)$ multiplications and N divisions. Consider the situation where we want to have a continuously tunable delay element using a Thiran fractional-delay filter. There are two options. Either, the coefficients of the intermediate filters are calculated during run-time as required, or they are obtained by interpolation. The former option is not desirable as the computational complexity of

Manuscript received May 16, 2006; revised 19 December 2006. The associate editor coordinating the review of this paper and approving it for publication was Dr. Dr Zoran Cvetković. This work was funded by the Engineering and Physical Sciences Research Council (EPSRC) Research Grant GR/S72320/01 Portfolio Partnership Award in Integrated Electronics.

The authors are with the I-Lab, Centre for Communication Systems Research (CCSR), University of Surrey, Guildford, Surrey, GU2 7XH, United Kingdom (e-mail: h.hacihabiboglu@surrey.ac.uk; b.gunel@surrey.ac.uk; a.kondoç@surrey.ac.uk)

Publisher Item Identifier XX.XXXX/XXXXX.XXXX.XXXXXX

calculating a new set of coefficients is high for high-order Thiran filters. The latter option, on the other hand, requires a computationally efficient algorithm for the derivation of a new allpass filter that approximates an intermediate fractional delay. Previously, low-order polynomial approximations to obtain the filter coefficient for fractional-delay filters, and corresponding variable filter structures have been proposed to solve the tunability problem [13], [14]. Although quite accurate, thanks to the direct algorithmic implementation of the approximation, such structures are computationally more complex than a simple IIR filter of the same order.

A root displacement method for obtaining intermediate fractional-delay filters was previously proposed by the authors [15]. The method paralleled similar approaches involved in LPC-based speech synthesis, namely for the interpolation of filter magnitude responses by pole shifting [16], or for speaker identity modification [17]. A similar method was also applied by the authors to the magnitude response interpolation of minimum-phase FIR filters used in binaural synthesis for 3D audio reproduction [18].

This article presents the analysis of the root displacement method with respect to its application in tunable fractional-delay filters and to the transient errors involved. The root displacement approach is summarized in Section II. Section III explains the effect of interpolation regarding the transient error due to coefficient update. Section IV presents a practical example of the utility of the proposed method in the physical modelling domain. Section V concludes the article.

II. INTERPOLATION OF THIRAN FILTERS USING ROOT DISPLACEMENT

Assume that we have two N^{th} -order Thiran fractional-delay filters, $H_1(z)$ and $H_2(z)$ which approximate two different fractional delays D_1 and D_2 . The aim of interpolation is to obtain a new allpass fractional-delay filter, $H_{\text{int}}(z)$, whose phase delay approximates an intermediate delay, D_{int} , such that $D_1 \leq D_{\text{int}} \leq D_2$. The following section provides an overview of the interpolation method previously proposed by the authors [15].

A. Root displacement method

An allpass IIR filter has reciprocal poles and zeros. The numerator polynomial of an allpass filter can be obtained simply by mirroring the order of the polynomial coefficients. Therefore, we can limit the interpolation process to the denominator polynomial, as the stability conditions for an IIR filter are imposed on the radii of the filter poles and as the numerator roots can be obtained trivially. The roots of a Thiran filter do not reveal themselves to a simple closed-form representation for $N \geq 5$ (Abel-Ruffini theorem). However, the roots can be obtained numerically by using the iterative polynomial root-finding techniques, by calculating the eigenvalues of the companion matrix by QR decomposition or by utilizing the Newton-Raphson method [19]. Fig. 1(a) shows the loci of the poles of 10^{th} -order Thiran fractional-delay filters modelling a fractional delay of $10 + d$ where $d \geq 0$. The poles of a Thiran filter approximating a fractional delay of 10.7 samples

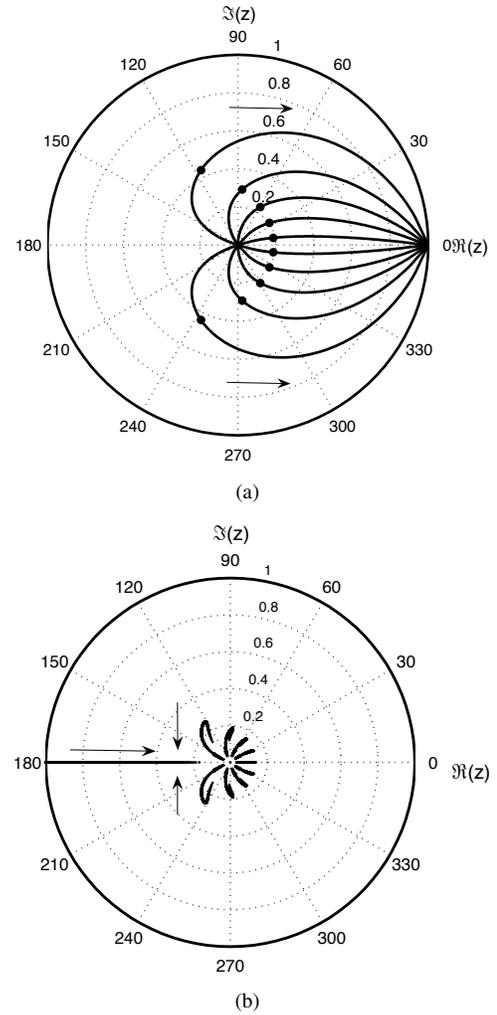


Fig. 1. The loci of the poles of 10^{th} -order Thiran filters approximating a fractional delay of $10 + d$ samples for (a) $d \geq 0$, and (b) $-1 < d < 0$. The directions of the arrows show increasing delay.

are denoted as filled circles. Fig. 1(b) shows the loci of the poles of 10^{th} -order Thiran fractional-delay filters modelling a fractional delay of $10 + d$ where $-1 < d < 0$. Considering that the filter has M_R real poles and M_C complex poles, it is possible to express the denominator polynomial, $A(z)$ in (1) in the following way:

$$A_i(z) = \prod_{k=1}^{M_R} (1 - r_{i,k} z^{-1}) \prod_{k=1}^{M_C/2} (1 - c_{i,k} z^{-1})(1 - c_{i,k}^* z^{-1}), \quad (5)$$

where $\{c_{i,k}, c_{i,k}^*\}$ is the k^{th} complex conjugate pole pair and $r_{i,k}$ are the real poles of the filter H_i .

The positive complex conjugate poles of filters $H_1(z)$ and $H_2(z)$ are first sorted with respect to their angles, and then paired according to their angular proximity. The interpolated complex poles can be calculated from the paired poles using:

$$c_{\text{int},k} = [1 - \rho] c_{1,k} + \rho c_{2,k}, \quad (6)$$

where ρ is a constant between 0 and 1. It was shown that the resulting interpolated Thiran filter is always stable as long as $0 \leq \rho \leq 1$ and the original filters are stable.

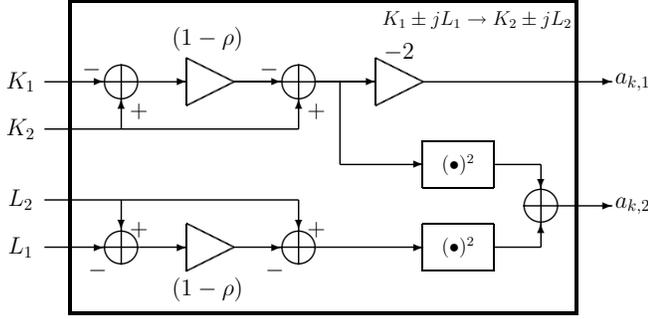


Fig. 2. Obtaining the filter coefficients of the interpolated second-order IIR section from original filter poles. The first coefficient $a_{k,0} = 1$. The total number of multiplications is 5. (From [15])

Consider two poles inside the unit circle, $c_{1,k} = K_1 + jL_1$ and $c_{2,k} = K_2 + jL_2$, with their complex conjugate pairs. The interpolated complex conjugate pole pair can be calculated as:

$$\begin{aligned} \{c_{int,k}, c_{int,k}^*\} &= K_{int,k} \pm jL_{int,k}, \\ &= [(1-\rho)K_1 + \rho K_2] \pm j[(1-\rho)L_1 + \rho L_2]. \end{aligned} \quad (7)$$

Due to the fact that the pole loci are not linear in pole angle, the relation between ρ and the phase delay is not exactly but only approximately linear. However for a small delay interval between the original filters, it can be considered as linear. The interpolated poles constitute the roots of the denominator polynomial, $A_{int,k}(z)$, of a second-order section of the allpass fractional-delay filter, $H_{int,k}(z) = B_{int,k}(z)/A_{int,k}(z)$, such that:

$$\begin{aligned} A_{int,k}(z) &= 1 + a_{k,1}z^{-1} + a_{k,2}z^{-2}, \\ &= 1 - 2K_{int,k}z^{-1} + (K_{int,k}^2 + L_{int,k}^2)z^{-2}. \end{aligned} \quad (8)$$

Thus, the denominator coefficients of a second-order section can be obtained easily by 4 multiplications and a shift operation from the real and complex parts of the original pole (see Fig. 2). Each real pole is represented by a first-order polynomial and thus a first-order section. Therefore, only two multiplications and a shift operation are required for interpolating a real pole. There are two options after this stage to obtain the overall filter:

- 1) If a direct-form implementation of the filter is sought for, the denominator polynomials are multiplied (i.e. convolved) to obtain the direct-form coefficients. The numerator polynomial can then be obtained by mirroring the obtained denominator coefficients. This option is undesirable as it will require additional computational effort.
- 2) The interpolated fractional-delay filter can be implemented as a cascade of tunable second-order allpass IIR sections corresponding to complex poles and first-order sections corresponding to real poles. The coefficients of the cascade sections can be obtained directly and easily as explained above.

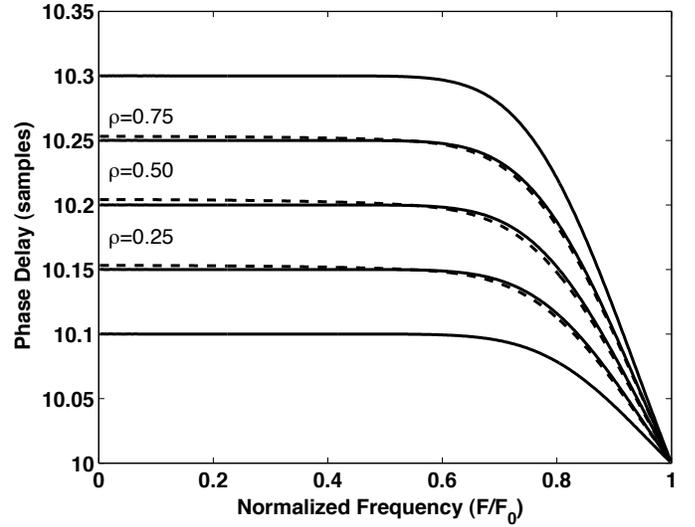


Fig. 3. Phase delays of Thiran (solid line) and interpolated (dashed line) fractional-delay filters. Interpolation is carried out using the Thiran filters approximating 10.1 and 10.3 samples delay. (From [15])

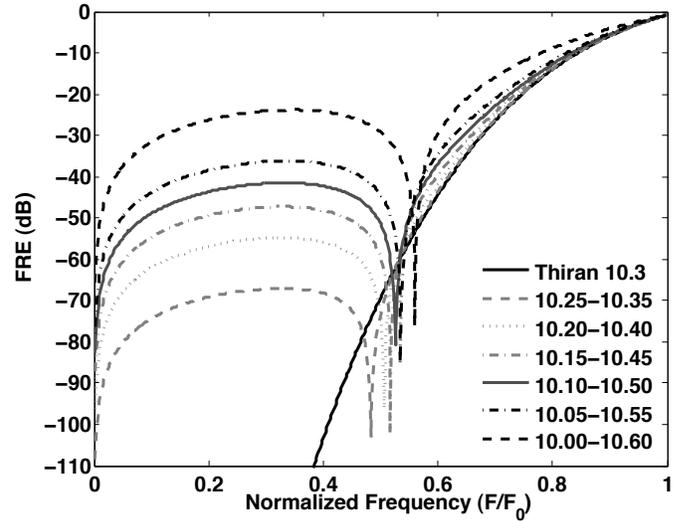


Fig. 4. Frequency response error (FRE) for interpolated allpass fractional-delay filters approximating 10.3 samples delay. Interpolation has been carried out using various pairs of Thiran filters.

Fig. 3 shows the phase delays of interpolated filters obtained from two 10^{th} -order Thiran fractional-delay filters modelling 10.1 and 10.3 samples delay. The figure depicts phase delays of the interpolated filters, as well as the phase delays of the calculated Thiran fractional-delay filters approximating the same delays. It may be observed that in comparison with the original filters representing the same fractional delay, the interpolated filters have a shorter bandwidth of flat phase delay. Nonetheless, the phase delays are in good agreement.

The absolute deviation of the complex-valued frequency response from the ideal frequency response can be quantified by the frequency response error, $FRE(\omega) = 20 \log |e^{-j\omega D} - H(\omega)|$. Fig. 4 shows the FRE functions for an interpolated filter approximating 10.3 samples delay ob-

tained using a variety of different original filters as well as the FRE function of the original Thiran filter approximating 10.3 samples delay. It may be observed that the error increases with the difference between the delays approximated by the original filters.

It should be noted that the proposed method does not allow interpolation when fractional parts, d , of the delays approximated by the two original filters have different signs. For example, it is not possible to utilize the proposed method to obtain an interpolated filter from two 10th-order Thiran filters approximating 9.9 ($d = -0.1$) and 10.3 ($d = 0.3$). This is due to the fact that the root loci for negative and positive values of d are very different as may also be observed from Fig. 1.

B. Comparison with linear interpolation

Using the two original filters, $H_1(z)$ and $H_2(z)$, the linearly interpolated filter $H_{lin}(z)$ can be obtained by a weighted average of the two transfer functions, such that:

$$H_{lin}(z) = \rho H_1(z) + (1 - \rho) H_2(z). \quad (9)$$

It is not possible to obtain an interpolated IIR filter from two IIR filters without increasing the filter order unless all the poles of the two original filters are common. For the general case where all the poles of the original filters are different, an interpolated filter can only be obtained at the cost of doubling the filter order. Therefore, given that the poles of two different Thiran filters cannot be common unless the two filters are the same, linear interpolation of the transfer functions is not a suitable option if the computational complexity and thus the memory requirements are not to be increased.

Another approach is to obtain the numerator and the denominator coefficients of a new filter by linear interpolation. With the aim of demonstrating the effect of linearly interpolating filter coefficients, let us consider the interpolated filter, $H_{lin}(z)$, obtained by calculating a new set of filter coefficients by weighted averaging from two sets of original coefficients obtained using the Thiran approximation, such that:

$$H_{lin}(z) = \frac{B_{lin}(z)}{A_{lin}(z)} = \frac{\sum_{k=0}^N [\rho a_{N-k} + (1 - \rho) b_{N-k}] z^{-k}}{\sum_{k=0}^N [\rho a_k + (1 - \rho) b_k] z^{-k}}, \quad (10)$$

where $\{a_k\}$ and $\{b_k\}$ represent the coefficients of the first and the second original filters, $H_1(z)$ and $H_2(z)$, respectively. The obtained coefficients may directly be used in the direct-form implementation of the filter. Although this interpolation method has the same computational complexity as the root-displacement interpolation, the obtained phase delay provides a worse interpolation accuracy as the filter coefficients do not have a linear relationship with the phase delay. Further, as will be shown in the subsequent sections, linear interpolation is less advantageous as the direct-form implementation causes higher transient errors when coefficients are changed.

Fig. 5(a) shows the phase delays of Thiran fractional-delay filters modelling the fractional delays of 16.1, 16.25, and 16.4 samples, as well as the interpolated filters modelling a fractional delay of 16.25 samples obtained from the given

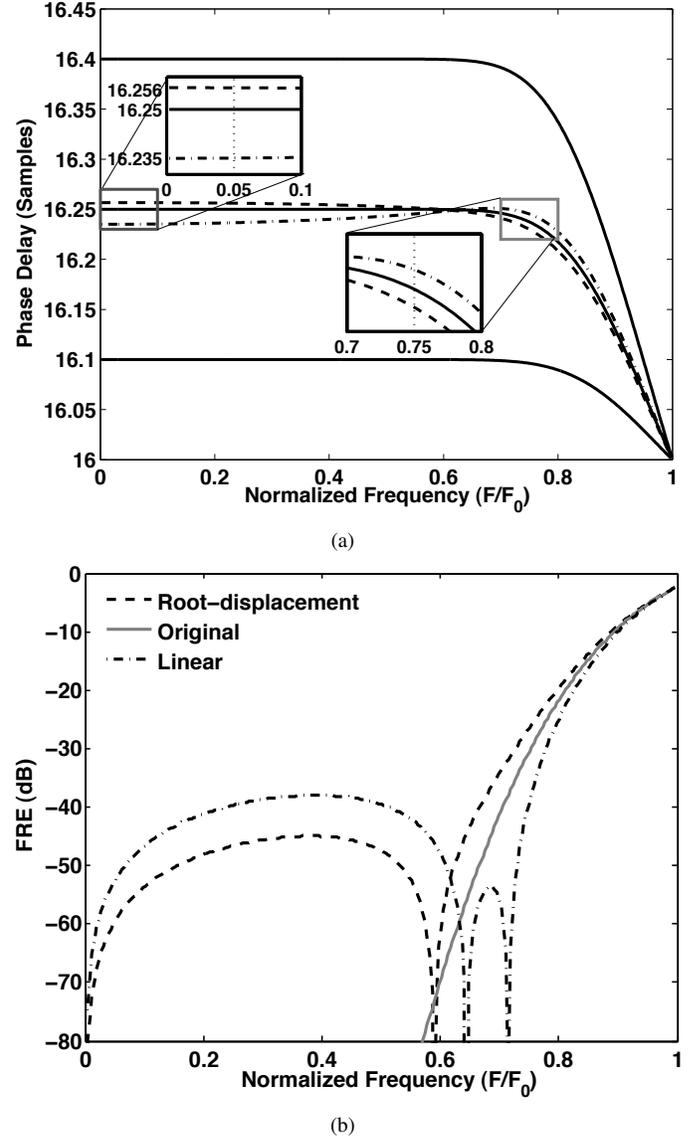


Fig. 5. (a) Phase delay of the original (solid line), linearly interpolated (dash-dot line), and root displacement interpolated (dashed line) fractional delay filters. The original filters model the fractional delays of 16.1, 16.25, and 16.4 samples. The original filters are Thiran filters modelling 16.1-sample and 16.4-sample fractional-delays, and (b) the corresponding frequency response errors (FREs).

Thiran filters using linear interpolation and root displacement interpolation. As may be observed, the absolute deviation of the phase delay of linearly interpolated filter is higher than that of the filter obtained using root displacement. In addition, the root displacement interpolation provides a phase delay which is flatter than the phase delay of the linearly interpolated filters. Fig. 5(b) shows the frequency response error (FRE) for the same filters. It may also be observed that the FRE is also higher for the linearly interpolated filter especially at the lower frequencies where the fractional-delay filter is required to approximate the desired fractional-delay well.

III. INTERPOLATION, TUNABILITY, AND TRANSIENTS

The root displacement interpolation method is computationally efficient when used in the calculation of a new set of filter coefficients that approximate an arbitrary delay between those approximated by two original filters. This allows using the method for designing computationally efficient tunable allpass fractional-delay filters.

However, one of the major implications of root displacement interpolation is that the filter is implemented in cascade form so as to avoid the unnecessary task of convolution while obtaining the coefficients. As with all recursive filters, changing the coefficients results in transient errors which are undesirable in many applications such as audio and acoustical signal processing. An analysis of transient errors occurring in direct-form II IIR filters and the application of the results to Thiran filters for direct-form and cascade implementations are given in this section.

A. Effective length of an IIR filter

The duration of the transient error is intricately related to the *effective length* of the impulse response of the IIR filter [20]. The effective length of the impulse response of an IIR filter can be quantified based on the cumulative energy of the impulse response at a given instant. Laakso and Välimäki [20] propose a method of calculating the effective length based on the explicit calculation of the impulse response for low order IIR filters and present a closed-form maximum boundary for higher-order IIR filters. The exact effective length of a filter's impulse response is defined as the instant $n = N_L$ at which the cumulative energy, $E_L(n) = \sum_{k=-\infty}^n h(k)^2$, of the impulse response, $h(n)$, is at a given percentage, P , of the total energy of the impulse response, $E = \sum_{k=-\infty}^{\infty} h(k)^2$, such that $E_L(N_L) = EP/100$. The notion of effective length as used in this paper refers to the original definition by Laakso and Välimäki [20], where the effective length is calculated directly from the original formula involving the impulse response.

B. Transient errors in direct-form II allpass IIR filters

1) *Analysis of the transient error:* When the coefficients of an IIR filter are changed at an arbitrary instant during runtime, transients will occur at the filter output. The total energy contained within the transient response can be reduced by selecting a suitable interpolation range [21]. This selection determines the total number of filter poles that need to be stored in look-up tables for a given application. As suggested previously [21], [22] the state-variable representation of an allpass IIR filter in direct-form II (DFII) structure [23], [24] can be used to model how the transient errors occur. Consider the following state-variable representation:

$$\mathbf{v}(n+1) = \mathbf{F}\mathbf{v}(n) + \mathbf{q}x(n), \quad (11)$$

$$y(n) = \mathbf{g}^T \mathbf{v}(n) + \mathbf{g}_0 x(n), \quad (12)$$

where,

$$\mathbf{v}(n) = [v(n) \ v(n-1) \ \dots \ v(n-N+1)]^T, \quad (13)$$

is the state vector at the instant n , $x(n)$ is the filter input, $y(n)$ is the filter output, and

$$\mathbf{F} = \begin{bmatrix} -a_1 - a_2 \dots - a_{N-1} - a_N \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 0 \end{bmatrix}, \quad \mathbf{q} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix},$$

$$\mathbf{g} = \begin{bmatrix} a_{N-1} - a_N a_1 \\ a_{N-2} - a_N a_2 \\ \vdots \\ a_1 - a_N a_{N-1} \\ 1 - a_N^2 \end{bmatrix}, \quad \mathbf{g}_0 = a_N. \quad (14)$$

Assume that the filter coefficients $\{a_k\}$ represented by the feedback matrix \mathbf{F} are changed to a new set of filter coefficients $\{b_k\}$ represented by the feedback matrix \mathbf{G} at an arbitrary instant, N_c , such that:

$$\mathbf{v}(n) = \begin{cases} \sum_{k=0}^{n-1} \mathbf{F}^{n-k-1} \mathbf{q}x(k) + \mathbf{F}^n \mathbf{v}(0), & 0 \leq n \leq N_c \\ \sum_{k=N_c}^{n-1} \mathbf{G}^{n-k-1} \mathbf{q}x(k) + \mathbf{G}^{n-N_c} \mathbf{v}(N_c), & n > N_c \end{cases} \quad (15)$$

In order to find the state vector difference, $\Delta \mathbf{v}(n)$, assume that we have two state vectors $\mathbf{v}_1(n)$ and $\mathbf{v}_2(n)$, where $\mathbf{v}_1(n)$ is defined as in (15), and $\mathbf{v}_2(n) = \sum_{k=0}^{n-1} \mathbf{G}^{n-k-1} \mathbf{q}x(k) + \mathbf{G}^n \mathbf{v}_2(0)$ representing the ideal state vector after the coefficient update. Neglecting the contribution of $\mathbf{v}_1(0)$, and $\mathbf{v}_2(0)$ we can define the state vector difference as $\Delta \mathbf{v}(n) = \mathbf{v}_1(n) - \mathbf{v}_2(n)$, such that:

$$\Delta \mathbf{v}(n) = \begin{cases} \sum_{k=0}^{n-1} (\mathbf{F}^{n-k-1} - \mathbf{G}^{n-k-1}) \mathbf{q}x(k), & 0 \leq n \leq N_c \\ \mathbf{G}^{n-N_c} \Delta \mathbf{v}(N_c). & n > N_c \end{cases} \quad (16)$$

Using (12) and (16) for $n \geq N_c$ it is possible to express the state vector difference and the transient error in the following state-variable representation:

$$\Delta \mathbf{v}(n+1) = \mathbf{G} \Delta \mathbf{v}(n), \quad n \geq N_c \quad (17)$$

$$\Delta y(n) = \mathbf{h}^T \Delta \mathbf{v}(n), \quad n \geq N_c \quad (18)$$

where $\mathbf{h}^T = [b_{N-1} - b_N b_1 \quad b_{N-2} - b_N b_2 \quad \dots \quad 1 - b_N^2]$, and $\Delta \mathbf{v}(n) = [\Delta v(n) \ \Delta v(n-1) \ \dots \ \Delta v(n-N+1)]^T$. Therefore, the transient error, $\Delta y(n)$ occurring at the filter output due to the coefficient change can be represented as the response of the second filter to the state-vector difference observed when the filter input, $x(n) = 0$ for $n \geq N_c$. It may then be suggested that the total energy of the transient error depends on $\Delta \mathbf{v}(N_c)$.

2) *State vector difference and the input signal:* It is possible to represent the state vectors $\mathbf{v}_1(n)$ and $\mathbf{v}_2(n)$ in the z -domain using the relation in (11) such that:

$$\mathbf{V}_1(z) = (\mathbf{I} - z^{-1} \mathbf{F})^{-1} \mathbf{q} X(z) z^{-1} = \mathbf{M}_1(z) X(z), \quad (19)$$

$$\mathbf{V}_2(z) = (\mathbf{I} - z^{-1} \mathbf{G})^{-1} \mathbf{q} X(z) z^{-1} = \mathbf{M}_2(z) X(z). \quad (20)$$

It can be shown that:

$$\mathbf{M}_1(z) = \frac{1}{\sum_{k=0}^N a_k z^{-k}} [z^{-1} z^{-2} \dots z^{-N}]^T, \quad (21)$$

$$\mathbf{M}_2(z) = \frac{1}{\sum_{k=0}^N b_k z^{-k}} [z^{-1} z^{-2} \dots z^{-N}]^T. \quad (22)$$

The state vector difference can then be represented in the z -domain as:

$$\begin{aligned} \Delta \mathbf{V}(z) &= [\mathbf{M}_1(z) - \mathbf{M}_2(z)] X(z), \quad (23) \\ &= \mathcal{M}_\Delta(z) [X(z) z^{-1} X(z) z^{-2} \dots X(z) z^{-N}]^T \quad (24) \end{aligned}$$

where $\mathcal{M}_\Delta(z)$ represents the recursive transfer function which relates the state vector difference to the input signal such as:

$$\mathcal{M}_\Delta(z) = \frac{\sum_{k=0}^N (b_k - a_k) z^{-k}}{\left(\sum_{k=0}^N a_k z^{-k}\right) \left(\sum_{k=0}^N b_k z^{-k}\right)}. \quad (25)$$

It is then possible to express in time domain the elements, $\Delta v(n-i)$ of the state vector $\Delta \mathbf{v}(n)$ as a convolution sum:

$$\begin{aligned} \Delta v(n-i) &= m_\Delta(n) * x(n-i-1) \\ &= \sum_{k=-\infty}^{\infty} x(n-i-k-1) m_\Delta(k), \quad (26) \end{aligned}$$

where $m_\Delta(n)$ is the impulse response associated with the transfer function, $\mathcal{M}_\Delta(z)$. It should be noted that due to causality, $m_\Delta(n) = 0$ for $n < 0$. For practical purposes it may also be assumed that $m_\Delta(n) \approx 0$ for $n > N_\Delta$ where N_Δ is the effective length of the given impulse response. It is then possible to write (26) as a finite sum such that:

$$\Delta v(n-i) = \sum_{k=0}^{N_\Delta-1} x(n-i-k-1) m_\Delta(k). \quad (27)$$

3) *Energy of the transient error:* Let us assume that the effective length of the second IIR filter and also the effective duration of the transient error are equal to N_L . It is then possible to define the truncated transient error in vectorial form using (17) and (18) as:

$$\overline{\Delta \mathbf{y}} = \mathcal{H} \overline{\Delta \mathbf{v}}, \quad (28)$$

where $\overline{\Delta \mathbf{y}} = [\Delta y(N_c) \Delta y(N_c+1) \dots \Delta y(N_c+N_L-1)]^T$ is the truncated transient error vector, $\mathcal{H} = [\mathbf{h} \mathbf{G}^T \mathbf{h} \dots (\mathbf{G}^{N_L-1})^T \mathbf{h}]^T$ is a N_L -by- N matrix, and $\overline{\Delta \mathbf{v}} = [\Delta v(N_c) \Delta v(N_c-1) \dots \Delta v(N_c-N+1)]^T$ is the state-vector difference. The energy of the transient error can be represented as:

$$E_{\Delta y} = \|\Delta y\|_F^2 = \|\mathcal{H} \overline{\Delta \mathbf{v}}\|_F^2. \quad (29)$$

Here $\|\bullet\|_F$ represents the Frobenius norm defined for a real-valued N -by- M matrix $\mathbf{X} = \{x_{i,j}\}$ as $\|\mathbf{X}\|_F = \sqrt{\text{Tr}(\mathbf{X}\mathbf{X}^T)} = (\sum_{i=1}^N \sum_{j=1}^M |x_{i,j}|^2)^{1/2}$ where $\text{Tr}(\bullet)$ represents the trace of a matrix. This expression gives an exact value of the total transient error energy within the effective length. However, as $\overline{\Delta \mathbf{v}} = \Delta \mathbf{v}(N_c)$ depends on the previous values of the input, only an upper bound can be given for the transient error energy unless the input signal is known.

Therefore, in what follows, the expression for the upper bound of the transient error energy is derived.

By definition, the following inequality holds for the Frobenius norm:

$$\|\mathcal{H} \overline{\Delta \mathbf{v}}\|_F \leq \|\mathcal{H}\|_F \|\overline{\Delta \mathbf{v}}\|_F. \quad (30)$$

Therefore, an upper bound for the transient energy can be defined as:

$$E_{\Delta y} \leq \|\mathcal{H}\|_F^2 \|\overline{\Delta \mathbf{v}}\|_F^2. \quad (31)$$

The squared value of each element of the state-difference vector given in (27) can be expressed as:

$$|\Delta v(n-i)|^2 = \left[\sum_{k=0}^{N_\Delta-1} x(n-i-k-1) m_\Delta(k) \right]^2. \quad (32)$$

An upper bound for this value exists due to the Cauchy-Schwarz inequality such that:

$$|\Delta v(n-i)|^2 \leq \sum_{k=0}^{N_\Delta-1} |m_\Delta(k)|^2 \cdot \sum_{k=0}^{N_\Delta-1} |x(n-i-k-1)|^2 \quad (33)$$

The second term of the right-hand side of (31) then obeys the following inequality:

$$\begin{aligned} \|\overline{\Delta \mathbf{v}}\|_F^2 &= \sum_{i=0}^{N-1} |\Delta v(N_c-i)|^2 \\ &\leq E_{m_\Delta} \sum_{i=0}^{N-1} \sum_{k=0}^{N_\Delta-1} |x(N_c-i-k-1)|^2, \quad (34) \end{aligned}$$

where $E_{m_\Delta} = \sum_{k=0}^{N_\Delta-1} |m_\Delta(k)|^2$ is the energy of the truncated impulse response of the difference transfer function, $\mathcal{M}_\Delta(z)$. The first term of the right hand side of the inequality in (31) can be expressed as:

$$\|\mathcal{H}\|_F^2 = \text{Tr}(\mathcal{H}\mathcal{H}^T) = \sum_{k=0}^{N_L-1} \mathbf{h}^T \mathbf{G}^k (\mathbf{G}^k)^T \mathbf{h}, \quad (35)$$

which is independent of the input and which only depends on the coefficients of the second filter that can be expressed as a constant such that $\|\mathcal{H}\|_F^2 = C$. It is then possible to express the upper bound for the transient energy as:

$$E_{\Delta y} \leq C E_{m_\Delta} \sum_{i=0}^{N-1} \sum_{k=0}^{N_\Delta-1} |x(N_c-i-k-1)|^2. \quad (36)$$

Thus the upper bound depends on the past $N_\Delta + N - 1$ samples of the input. Given that the magnitude of the input signal is bounded (i.e. $-1 \leq x(n) \leq 1$), the absolute maxima of the last term above will be obtained if $|x(n)|^2 = 1$ for $N_c - 1 \leq n \leq N_c - (N_\Delta + N - 1)$. It is thus possible to suggest that the following upper bound exists for the transient error energy regardless of the input signal:

$$E_{\Delta y} \leq C E_{m_\Delta} N N_\Delta. \quad (37)$$

The upper bound for the RMS transient error is then defined as:

$$\Delta y_{\text{rms}} = \sqrt{\frac{E_{\Delta y}}{N_L}} \leq \sqrt{\frac{C E_{m_\Delta} N N_\Delta}{N_L}}. \quad (38)$$

$$\mathcal{M}_\Delta(z) = \frac{{}_2F_1(-N, d_a + \tau; d_a + \tau + N + 1; z^{-1}) - {}_2F_1(-N, d_a; d_a + N + 1; z^{-1})}{{}_2F_1(-N, d_a; d_a + N + 1; z^{-1}) {}_2F_1(-N, d_a; d_a + N + \tau + 1; z^{-1})}. \quad (40)$$

4) *Coefficient update rate*: Changing filter coefficients before the transient error has effectively vanished will result in the overlap and thus the build-up of transient errors and eventually increase the RMS transient error energy observed at the output. This means that the coefficient update rate should be chosen according to the effective length of the transient error. Therefore, it is suggested that the maximum coefficient update rate is defined as the inverse of the effective length of the recursive filter, i.e. $f_{\text{update}} = 1/N_L$. This way, the upper bound of the RMS transient error will not be exceeded. The coefficients of a filter with a higher effective length will therefore have to be updated less often than those of a filter with a lower effective length, reducing the maximum allowable rate of coefficient update.

5) *Remarks*: The following general conclusions may be drawn from the discussions in this section:

- 1) The transient error is not related to input samples, $x(n)$, after the coefficients are updated (i.e. $n \geq N_c$), but those prior to this update. The total transient energy is solely related to the state-vector difference, $\Delta \mathbf{v}(n)$ at $n = N_c$. In other words, keeping the state vector difference, $\Delta \mathbf{v}(N_c)$ small will result in a small transient error energy. This in turn requires that the change in the filter coefficients is not big.
- 2) The duration of the transient error is related to the response of the updated filter to the discrepancy at the filter state vector. However, the effective length of the impulse response of the IIR filter is still considered to be a measure of the duration of the transient error [20].
- 3) The upper bound of the total transient energy is related to the length of the filter, N , and the effective length of the impulse response of the difference transfer function, $\mathcal{M}_\Delta(n)$ which is also longer for greater N . Therefore, a lower-order filter having all of its poles common with a higher-order filter will have a smaller upper bound for the energy of the transient error than the higher order filter.

C. Transient errors and DFII Thiran fractional-delay filters

As originally shown by Thiran [11], it is possible to express the denominator polynomial of a maximally-flat group-delay filter as a truncated Gauss hypergeometric series, ${}_2F_1(a, b; c; z)$ defined as:

$${}_2F_1(a, b; c; z) = \frac{\Gamma(c)}{\Gamma(a)\Gamma(b)} \sum_{n=0}^N \frac{\Gamma(a+n)\Gamma(b+n)}{\Gamma(c+n)} \frac{z^n}{n!}, \quad (39)$$

where $\Gamma(x)$ is the Gamma function [25].

Let us consider the two N^{th} -order original filters having the respective set of coefficients, $\{a_k\}$, and $\{b_k\}$, approximating delays $N + d_a$, and $N + d_b$, such that $d_b = d_a + \tau$. It is then possible to represent the state difference transfer function, $\mathcal{M}_\Delta(z)$ in closed form as in (40).

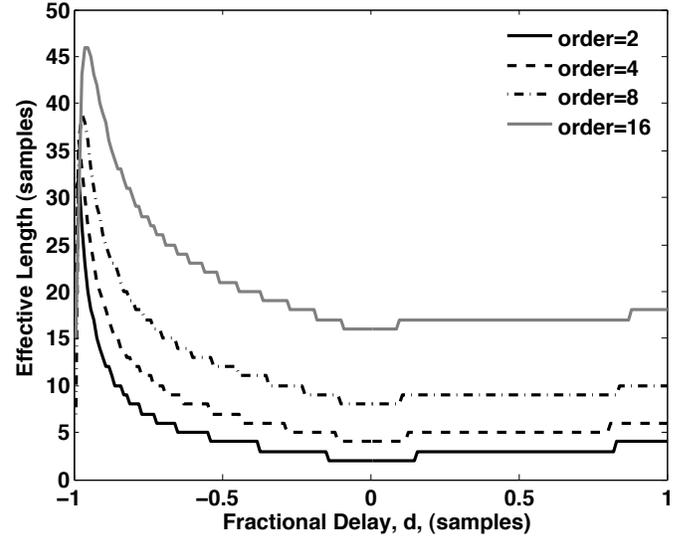


Fig. 6. The effective length ($P = 99\%$) of impulse responses for Thiran filters of order 2,4,8, and 16. For the fractional part of the delay from -1 to 1.

This transfer function is stable as the two multiplicative terms in the denominator both have roots corresponding to roots of the first and the second filters that are inside the unit circle.

Fig. 6 shows the effective length, N_L , of the impulse responses for $P = 99\%$ for various Thiran fractional-delay filters. It may be observed that the duration of the transient error increases with the absolute fractional delay. The figure also shows that, higher order Thiran fractional-delay filters will have longer transient error durations, and thus a higher number of output samples contaminated with the transient error.

Figure 7 shows the upper limit of the RMS transient error for a change in the coefficients of a DFII Thiran filter, modelling a delay of $(N + d)$ samples to those of one modelling $(N + d + 0.1)$ samples, where d is the fractional part of the delay and N is the filter order. It may be observed that the upper bound of the RMS transient error increases with the order of the Thiran filters. Considering that the effective length of a higher order Thiran filter is also higher, the total transient energy corresponding to a small change in the modeled delay will be high if the DFII implementation is used. This is another disadvantage of storing a small set of direct-form filter coefficients and obtaining the new coefficients by linear interpolation.

D. Transient error in cascade form implementation

As stated earlier, obtaining the coefficients of second-order IIR filters from interpolated roots is straightforward and computationally more efficient than calculating the direct-form polynomial coefficients from interpolated roots. Therefore, in

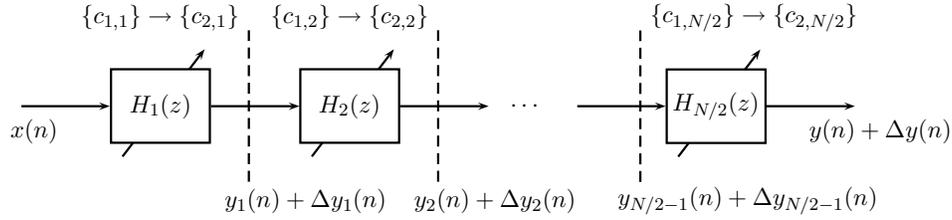


Fig. 8. Cascade implementation of an N^{th} -order (N even) filter. The transient error of each section is propagated through the subsequent sections.

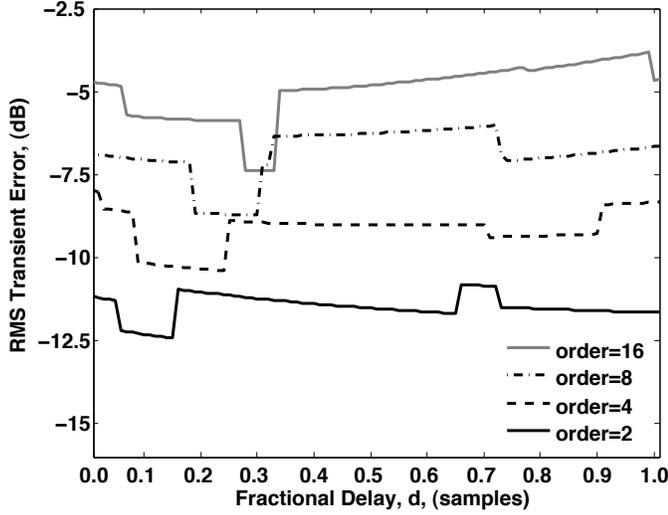


Fig. 7. The upper bounds of RMS transient errors in dB for Thiran filters of order $N = 2, 4, 8,$ and 16 for a change of coefficients modeling a delay of $(N + d)$ to $(N + d + 0.1)$ for the fractional part of the delay d between 0 and 1.

order to improve the computational efficiency of the proposed interpolation algorithm, the cascade form implementation is considered. As will be discussed shortly, the cascade implementation also reduces the level of the transient error.

Cascade implementation basically amounts to the decoupling of filter roots. For the case of Thiran filters, all cascaded sections can be selected to be allpass, so that the output of each allpass section is input to the next allpass section. Consider the cascade implementation of the N^{th} -order Thiran filter shown in Fig. 8. As there exists no global feedback and all of the IIR sections in the cascade chain are allpass, the transient error occurring at the output of each cascaded section due to coefficient switching only has an additive effect on the output of the cascade chain. Also, because the IIR sections are not zero-phase, transient error at the output of each second-order section experiences a non-zero delay effectively shifting each individual transient error in time.

Assume that the coefficients of all second order sections are updated at time $n = N_c$. In order to find an upper bound for the transient error energy let us first express the transient error observed at the output of a second order section, $\Delta y_i(n)$, in frequency domain as $\Delta Y_i(e^{j\omega})$, and the overall transient error, $\Delta y(n)$, in frequency domain as $\Delta Y(e^{j\omega})$. It can be shown by the triangle inequality that:

$$|\Delta Y(e^{j\omega})| = \left| \Delta Y_{N/2}(e^{j\omega}) + \sum_{i=1}^{N/2-1} \Delta Y_i(e^{j\omega}) e^{-j \sum_{k=i+1}^{N/2} \angle H_k(e^{j\omega})} \right| \leq \sum_{i=1}^{N/2} |\Delta Y_i(e^{j\omega})|. \quad (41)$$

Now consider the squared magnitude response. It is possible to define an upper bound for the squared magnitude spectrum of the transient error by using the Cauchy-Schwarz inequality such that:

$$|\Delta Y(e^{j\omega})|^2 \leq \left(\sum_{i=1}^{N/2} |\Delta Y_i(e^{j\omega})| \right)^2 \leq \frac{N}{2} \sum_{i=1}^{N/2} |\Delta Y_i(e^{j\omega})|^2. \quad (42)$$

By expressing the transient error energy in the frequency domain using the Parseval's relation, the upper bound of the overall transient energy can be expressed as:

$$E_{\Delta y} = \frac{1}{2\pi} \int_0^{2\pi} |\Delta Y(e^{j\omega})|^2 d\omega \leq \frac{N}{2} \sum_{i=1}^{N/2} \frac{1}{2\pi} \int_0^{2\pi} |\Delta Y_i(e^{j\omega})|^2 d\omega. \quad (43)$$

Equality will only hold if all transient error energies are equal, and if the transient errors completely overlap in time. This is not possible in practice as all the second-order sections have non-zero delays. In other words, the total transient error energy will always be smaller than $N/2$ times the sum of transient error energies, $E_{\Delta y_i}$, observed at the output of each second order section. On the other extreme, if the transient errors are fully non-overlapping in time, the overall transient error is the sum of individual transient error energies. This also is not possible in practice as the recursive filters have infinite duration impulse responses which will inevitably overlap in time. This provides a lower bound for the overall transient error energy such that:

$$\sum_{i=1}^{N/2} E_{\Delta y_i} < E_{\Delta y} < \frac{N}{2} \sum_{i=1}^{N/2} E_{\Delta y_i} \quad \text{for } N > 2. \quad (44)$$

The upper and lower bounds of the transient error energy at the output of each cascaded second-order section can be calculated as in (37). The overall effective length of the cascade

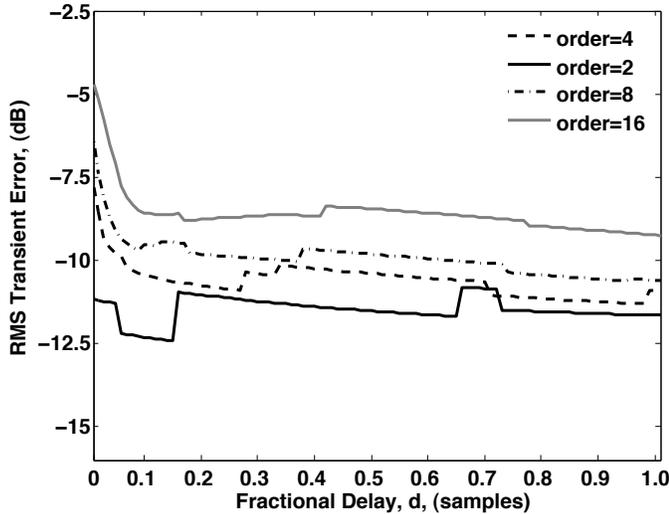


Fig. 9. The upper bounds of RMS transient errors in dB for Thiran filters in cascade implementation for orders $N = 2, 4, 8$, and 16 for a change of coefficients modeling a delay of $(N + d)$ to $(N + d + 0.1)$ for the fractional part of the delay d between 0 and 1.

implementation is the same as the direct implementation, and the bounds of the RMS transient error are defined accordingly by using the effective length of the overall impulse response of the cascade structure.

Figure 9 shows the upper bound of the RMS transient error for the same filter orders as analyzed for the direct implementation (see Fig. 7). The figure is drawn at the same scale so that a direct comparison is possible. It may be observed that the RMS transient error for the cascade implementation is around 3 dB below that of the same order direct implementation. Note that the RMS transient error for the DFII and cascade implementation are the same when the order, $N = 2$, as expected.

Fig. 10 shows the effect of switching the fractional delay from 10.1 to 10.5 samples at $N_c = 400$ for a sinusoidal input of $f = 320$ Hz at a sampling frequency of $F_0 = 8$ kHz. The first panel shows the ideal output together with the output of cascade and direct implementations. The second panel shows the error at the output of the direct implementation. The third panel shows the error at the output of the cascade structure. It may be observed that the cascade implementation reduces the level of the transient error considerably as discussed above.

IV. EXAMPLE: DIGITAL WAVEGUIDE STRING WITH FRACTIONAL DELAYS

Digital waveguides (DWG) present an efficient way to synthesise musical instrument sounds, model the vocal tract, or simulate room acoustics. A very simple example is the DWG string model [26] which is used in the modelling of plucked string instruments like the guitar. The DWG string is essentially a damped digital oscillator. The fundamental frequency of the model depends on the length of the digital string, or in other words the total delay in the feedback path. It is possible to use tunable fractional-delay filters [27], [28] to change the total delay in the feedback path to model

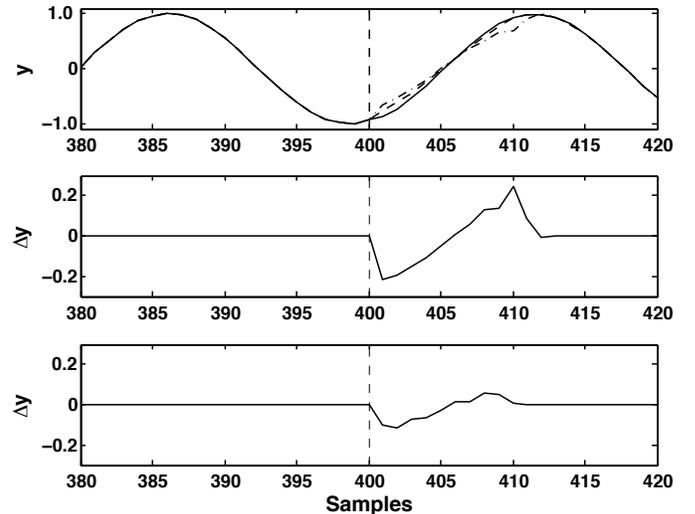


Fig. 10. The outputs and the transient errors of different implementations of a 10th-order IIR fractional delay filter for a 320 Hz sinusoidal input at $F_s = 8$ kHz. Coefficients are switched from those of 10.1 samples delay to 10.5 samples delay at the 400th sample. (a) The ideal output (solid line), the output of the the direct-form II implementation (dash-dot line), and the output of the cascade implementation (dashed line), (b) the transient error for the direct-form II implementation, (c) the transient error for the cascade structure.

effects like *glissando* (i.e. continuous pitch shift). Fig. 11 shows the digital waveguide string whose feedback path is implemented with a fixed delay of 10 samples, and five 4th-order Thiran fractional-delay filters in series. The filter $W_d(z) = 0.965/(1 - 0.03z^{-1})$ is a single pole loop filter for damping the feedback signal [28].

For demonstrating the effect of interpolation of the fractional-delay filters, we excite the digital waveguide string with a Hamming pulse of 10 samples width. Fig. 12(a) shows the output and the spectrogram of the “freely vibrating” DWG string when $\rho = 1$. The interpolation method proposed in this article is used to interpolate the fractional delay modelled by each Thiran filter (i.e. $4.0 + d$) between 8.0 and 4.0 samples with interpolation steps of 0.02 samples by slowly decreasing ρ . Initially, the total delay (i.e. $20.0 + 5d$) of the digital waveguide string is 50 samples corresponding to a fundamental frequency of $f = 160$ Hz at a sampling frequency of $F_0 = 8$ kHz. Coefficient update is carried out every 40 samples. At the 8000th sample, the total delay modelled by each fractional-delay filter is 4 samples, decreasing the total length of the digital waveguide string to 30 samples. This length corresponds to 266 Hz at the same sampling frequency. Fig. 12(b) shows the magnitude spectrogram of this glissando effect. Frequency shift may be observed both with the fundamental frequency and with the higher harmonics.

V. CONCLUSIONS

An analysis of the application of a simple and effective interpolation method for tunable allpass fractional-delay filters was presented in this article. It was shown that the proposed interpolation method results in a more accurate interpolation than the linear interpolation of filter coefficients. The transient

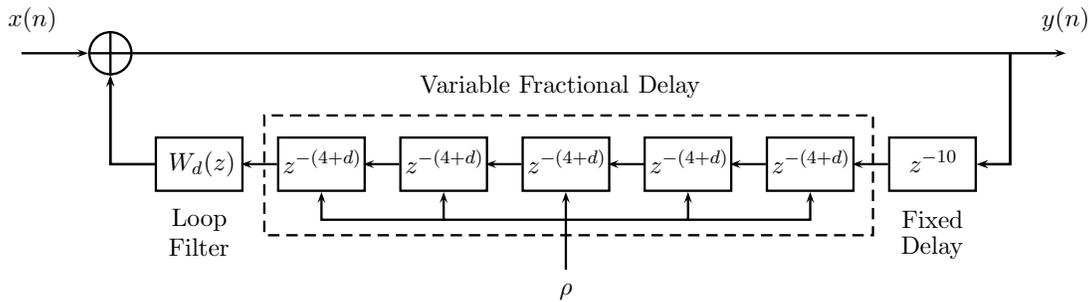


Fig. 11. Simple digital waveguide string as implemented with variable Thiran fractional-delay filters in the feedback path. The variable fractional-delay filters are of order 4. The delay is controlled by the interpolation parameter denoted as ρ . $W_d(z)$ represents the loop filter used to damp the feedback signal

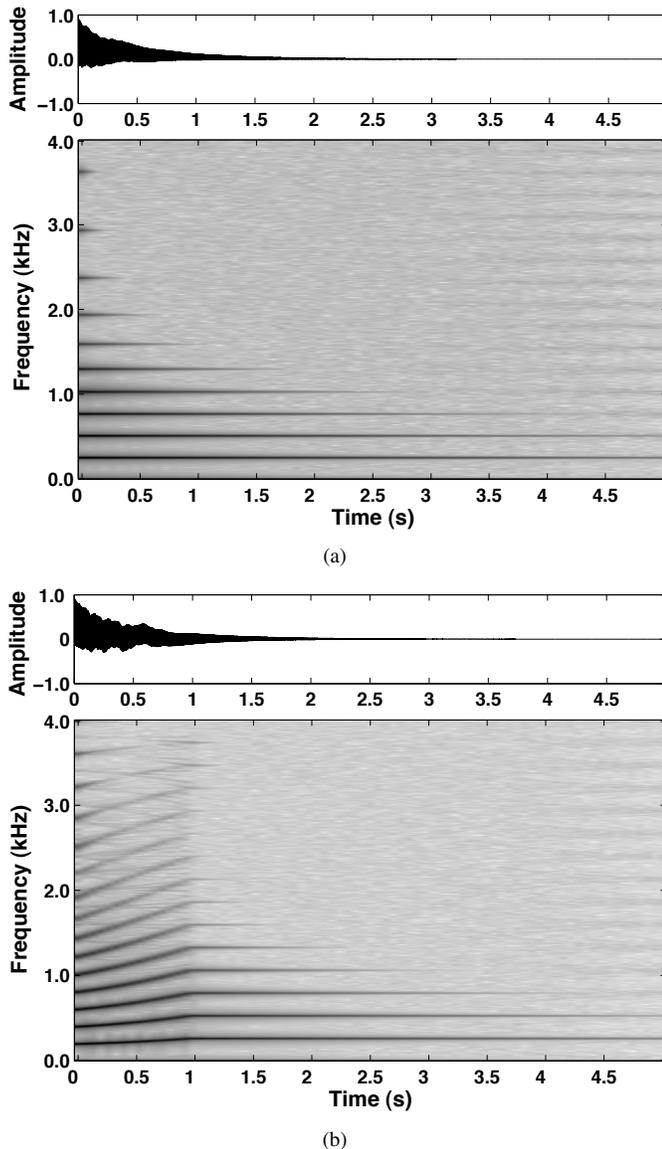


Fig. 12. Output signals and the magnitude spectrograms of (a) the “freely-vibrating” DWG string and (b) the frequency shift (i.e. glissando) on the digital waveguide string carried out by the proposed interpolation method.

errors in IIR filters that arise from coefficient update were analyzed. The upper bound for the RMS transient error,

independent of the filter input, was derived for a general IIR filter in DFII implementation. The maximum allowable coefficient update rate for which the RMS transient error upper bound is not exceeded was defined based on the effective length of the filter impulse response. The numerical bounds of RMS transient errors were obtained for several allpass fractional-delay filters in direct-form implementation. The upper and lower bounds of the RMS transient error for a cascade implementation consisting of DFII blocks was derived based on the transient error bounds for general DFII IIR filters. It was demonstrated that changing the filter coefficients of the cascade implementation results in lower transient error bounds in general. A practical example utilizing the proposed interpolation method was given, in which the glissando effect was synthesized on a digital waveguide string by modelling the feedback delay with a cascade of tunable fractional-delay filters interpolated by the proposed method. The proposed method has applications not only in physical model based sound synthesis and audio effects, but also in a variety of other applications including speech synthesis, microphone array processing, telecommunications systems, and spatial audio synthesis.

ACKNOWLEDGMENT

This paper is dedicated to the memory of the first author’s grandfather, Lütfü Sürmeli. The authors would like to thank the associate editor Dr. Zoran Cvetković and three anonymous reviewers for their feedback and insightful comments.

REFERENCES

- [1] T. I. Laakso, V. Välimäki, M. Karjalainen, and U. K. Laine, “Splitting the unit delay,” *IEEE Signal Processing Mag.*, vol. 13, no. 1, pp. 30–60, January 1996.
- [2] L. Savioja, J. Huopaniemi, T. Lokki, and R. Väänänen, “Creating interactive virtual acoustic environments,” *J. Audio Eng. Soc.*, vol. 47, no. 9, pp. 675–705, September 1999.
- [3] V. Välimäki, “Discrete-time modeling of acoustic tubes using fractional delay filters,” Ph.D. Thesis, Helsinki University of Technology, Espoo, Finland, December 1995.
- [4] V. Välimäki, M. Karjalainen, and T. Kuusma, “Articulatory speech synthesis based on fractional delay waveguide filters,” in *Proc. 1994 IEEE Int. Conf. Acoust. Speech, Signal Process. (ICASSP94)*, Adelaide, Australia, April 1994, pp. 1/585–1/588.
- [5] C. W. Farrow, “A continuously variable digital delay element,” in *Proc. 1988 IEEE Int. Symp. Circuits and Systems (ISCAS88)*, Espoo, Finland, June 1988, pp. 2641–2645.

- [6] K. Rajamani, Y.-S. Lai, and C. W. Farrow, "An efficient algorithm for sample rate conversion from CD to DAT," *IEEE Signal Processing Lett.*, vol. 7, no. 10, pp. 288–290, October 2000.
- [7] P. Murphy, A. Krukowski, and A. Tarczynski, "An efficient fractional sample delay for digital beam steering," in *Proc. 1997 IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP97)*, Munich, Germany, April 1997, pp. III/2245–III/2248.
- [8] E. Hermanowicz, "Explicit formulas for weighting coefficients of maximally flat tunable FIR delayers," *Elect. Lett.*, vol. 28, no. 20, pp. 1936–1937, September 1992.
- [9] E. Hermanowicz and M. Rojewski, "Design of FIR first order digital differentiators of variable fractional sample delay using maximally flat error criterion," *Elect. Lett.*, vol. 30, no. 1, pp. 17–18, January 1994.
- [10] A. Lee, M. Ahmadi, V. Ramachandran, and C. S. Gargour, "Design of fractional delay filters," *Comput. Electr. Eng.*, vol. 27, no. 3, pp. 287–292, May 2001.
- [11] J.-P. Thiran, "Recursive digital filters with maximally flat group delay," *IEEE Trans. Circuit Theory*, vol. CT-18, no. 6, pp. 659–664, November 1971.
- [12] A. Fettweis, "A simple design of maximally flat delay digital filters," *IEEE Trans. Audio Electroacoust.*, vol. AU-20, no. 2, pp. 112–114, June 1972.
- [13] M. Makundi, T. I. Laakso, and V. Välimäki, "Efficient tunable IIR and allpass filter structures," *Elect. Lett.*, vol. 37, no. 6, pp. 344–345, March 2001.
- [14] J.-S. Park, B.-K. Kim, J.-G. Chung, and K. K. Parhi, "High-speed tunable fractional-delay allpass filter structure," in *Proc. 2003 IEEE Int. Symp. Circuits and Systems (ISCAS03)*, Bangkok, Thailand, May 2003, pp. IV/165–IV/168.
- [15] H. Hacıhabiboğlu, B. Günel, and A. M. Kondoç, "Interpolated allpass fractional-delay filters using root displacement," in *Proc. 2006 IEEE Int. Conf. Acoust., Speech, Signal Process.*, Toulouse, France, May 2006, pp. III/864–III/867.
- [16] V. Goncharoff and M. Kaine-Krolak, "Interpolation of LPC spectra via pole shifting," in *Proc. 1995 IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP95)*, Detroit, MI, May 1995, pp. I/780–I/783.
- [17] J. Slifka and T. R. Anderson, "Speaker modification with LPC pole analysis," in *Proc. 1995 IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP95)*, Detroit, MI, USA, May 1995, pp. I/644–I/647.
- [18] H. Hacıhabiboğlu, B. Günel, and A. M. Kondoç, "Head-related transfer function filter interpolation by root displacement," in *Proc. 2005 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA05)*, New Paltz, NY, October 2005, pp. 134–137.
- [19] H. J. Orchard, "The roots of the maximally flat-delay polynomials," *IEEE Trans. Circuit Theory*, vol. CT-12 (Corresp.), pp. 452–454, September 1965.
- [20] T. I. Laakso and V. Välimäki, "Energy-based effective length of the impulse response of a recursive filter," *IEEE Trans. Instrum. Meas.*, vol. 48, no. 1, pp. 7–17, February 1999.
- [21] R. Rabenstein, "Minimization of transient signals in recursive time-varying filters," *Circuits, Syst., Signal Process.*, vol. 7, no. 3, pp. 345–359, 1988.
- [22] L. H. Zetterberg and Q. Zhang, "Elimination of transients in adaptive filters with application to speech coding," *Signal Processing*, vol. 15, no. 4, pp. 419–428, 1988.
- [23] D. Schlichthärle, *Digital Filters: Basics and Design*, 1st ed. London, UK: Springer Verlag, 2000.
- [24] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing: Principles, Algorithms, and Applications*, 3rd ed. New Jersey, USA: Prentice Hall, 1996.
- [25] M. Abramowitz and I. Stegun, *Handbook of Mathematical Functions*, 10th ed. Washington D.C., USA: U.S. Government Printing Office, 1972.
- [26] J. O. Smith, "Physical modelling using waveguides," *Comput. Music J.*, vol. 16, no. 4, pp. 75–87, 1992.
- [27] V. Välimäki and M. Karjalainen, "Implementation of fractional delay waveguide models using allpass filters," in *Proc. 1995 IEEE Int. Conf. Acoust., Speech, Signal Process.*, Detroit, MI, May 9–12 1995, pp. II/1524–II/1527.
- [28] 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 Conference*. Banff, Canada: International Computer Music Association, 1995, pp. 303–306.



Hüseyin Hacıhabiboğlu (S'96-M'00) was born in Ankara, Türkiye in 1978. He received his B.Sc. (Hons) in electrical and electronic engineering from Middle East Technical University in 2000, M.Sc. in electrical and electronic engineering from University of Bristol in 2001. He was awarded his Ph.D. in computer science from Queen's University Belfast in 2004 for his research into the simplification of signal processing algorithms used in room auralization by the application of psychoacoustical knowledge. Since 2004, he has been with the Multimedia and DSP Research Group (I-Lab) at the Centre for Communication Systems Research (CCSR), University of Surrey. His research interests include audio signal processing, room acoustics modelling and simulation, psychoacoustics of spatial hearing, and microphone array processing. He is one of the core members of the EPSRC-funded Noise Futures Network. Dr Hacıhabiboğlu is a member of the IEEE, Audio Engineering Society, and European Acoustics Association.



Banu Günel (S'97-M'00) was born in Antalya, Türkiye, in 1977. She received her B.Sc. degree in electrical and electronic engineering from the Middle East Technical University, Ankara, Türkiye in 2000 and M.Sc. degree in communication systems and signal processing from the University of Bristol, UK in 2001. In 2004, she completed her Ph.D. on audio and acoustical signal processing at the Queens University of Belfast, UK. Since 2004, she is working as a research fellow in Centre for Communication Systems Research (CCSR) at the University of Surrey, UK. Her main research interests are spatial audio, array signal processing, acoustic imaging, psychoacoustics and communication acoustics. She is a member of the European Acoustics Association (EAA), Audio Engineering Society (AES) and the IEEE.



Ahmet M. Kondoç (M'91) was born in Cyprus. He received the B.Sc. (Hons.) degree in engineering, the M.Sc. degree in telematics, and the Ph.D. in communication in 1983, 1984, and 1986, respectively. He became a Lecturer in 1988, a Reader in 1995, and then in 1996, a Professor in Multimedia Communication Systems and deputy director of Centre for Communication Systems Research (CCSR), University of Surrey, Guildford, U.K. He has over 250 publications, including a book on low-bit-rate speech coding and several book chapters. He has graduated more than 40 Ph.D. students in the areas of speech/image and signal processing and wireless multimedia communications, and has been a consultant for major wireless media terminal developers and manufacturers. Prof. Kondoç has been awarded several prizes, the most significant of which are The Royal Television Societies Communications Innovation Award and The IEE Benefactors Premium Award. He has been on the Refereeing College for EPSRC and on the Canadian Research Councils. He is a member of the IEEE and the IET.