RECURSIVE EXTENDED WINDOW INTERPOLATORS APPLIED TO THE DESIGN OF IIR AND FRACTIONAL DELAY FILTERS

G. I. Braileanu
Gonzaga University
Spokane, Washington 99258, USA
braileanu@gonzaga.edu

ABSTRACT

This paper proposes a conceptual modification of the interpolator used by the extended window design (EWD) of IIR and fractional delay filters. The original EWD digitizes analog filters by performing a trigonometric polynomial interpolation along an extended window that ends at the current sampling time. The proposed changes increase the flexibility and the degrees of freedom of the interpolator in two ways. First, the interpolation space is expanded from the space spanned by a restricted set of sinusoidal functions to the space spanned by any set of linearly independent functions. Second, the Hermite interpolation (which processes the values of both the signal and some of its derivatives at sampling times) is considered in order to smoothen the interpolated signal. Since the input derivatives are not available, related parameters are estimated and updated recursively. The increased accuracy of the proposed recursive EWD relatively to nonrecursive EWD and WLS optimal filter designs is illustrated with an example.

1. INTRODUCTION

A versatile method for the design of discrete-time equivalents of analog systems, termed extended window design (EWD), was mainly developed for fractional delay (FD) applications [1],[2]. The rationale for the digital filter design based on *analog prototypes* is twofold. First, since both the analog and digital transfer functions are analytical rational functions, the properties of such functions dramatically simplify the design problem relatively to the conventional iterative weighted least squares (WLS) IIR filter design which is meant to approximate arbitrary frequency responses. Unlike the WLS approach, the EWD provides a straightforward design with no need for iterations or trial-and-error adjustments of weighting functions. Second, in contrast to the filter design performed directly in the digital domain, the nature of the digitizing equations which approximate the output signals of analog filters leads to an efficient solution to the problem of implementing fractional delays by digital means [2] encountered, for example, in sampling rate conversion. Basically, the traditional two-point linear interpolation method for digitizing analog filters [3]-[5] – that yields the so-called triangle-hold equivalents – was modified in [1] in order to use (m+1)-point interpolators ($m \geq 2$). The digitizing errors produced by such fixed-length interpolators are due to the end segments which contain most of the error energy [6], and can be reduced by adding a small delay, or by increasing the filter order while the additional poles are placed at z=0. Indeed, these are two options where the approximation of the analog filter response is done by ignoring one or both end segments of the interpolated input. Thus, in contrast to the traditional digitizing methods [3]-[5] that use only the single time period that ends at the "current" sampling time, the EWD interpolation window is extended even beyond the time segment that is actually used to compute the response of the analog filter. Nevertheless, the error of the working middle segment of the interpolated input may still contain relatively large oscillations.

The method proposed in this paper is more general than the previously developed EWD in two ways. First, the interpolation space is expanded from the space spanned by a set of sinusoidal functions to the space spanned by any set of linearly independent functions. In particular, the latter may be the modes of the analog prototype. But the most distinctive feature is the smoothening of the left end of the interpolated segment whose impact on the output usually outweighs the effect of the most current values of the input. To this end, a modified Hermite interpolation [6] is considered in a form that requires the interpolated segment to satisfy a few additional conditions. Such conditions may be values of the derivatives at some sampling times[6], or signal values in-between the sampling times. Since these values are not actually available, related parameters are estimated and updated recursively.

The paper is organized as follows. Section 2 defines the new fixed-length interpolators in terms of the so-called characteristic functions that will be used later for design. Section 3 derives the closed form expression of the IIR transfer functions produced with the recursive extended window design (R-EWD). It is worth noting that this design is presented with reference to the original EWD for two reasons. First, this allows for a clear comparison between the two methods. Second, due to space limitations, only the theoretical derivation based on the partial fraction expansion will be presented in this paper. Yet, it will be possible to relate the above mentioned derivation to the numerically stable EWD algorithm presented in [1] in terms of linear algebra relationships and, finally, providing the option of FD implementations [2],[7]. Finally, the increased accuracy of the R-EWD relatively to EWD and WLS optimal filter designs is illustrated with an example in the concluding Section 4.

2. THE EWD AND R-EWD INTERPOLATORS

The R-EWD filters are designed below as digital equivalents of *analog prototypes* defined by transfer functions of the form

$$H(s) = \frac{Y(s)}{X(s)} = \frac{N(s)}{D(s)} = \frac{N(s)}{s^{n_A} + a_1 s^{n_A-1} + \ldots + a_{n_A}}, \quad (1)$$

where $N(s)$ and $D(s)$ are known polynomials of orders m_A and n_A, respectively. Without loss of generality, *the time is normalized to the sampling period*, and so the *folding frequency* will be $\omega_f = \pi$. In the following, the signals $x(t)$ and $y(t)$ are the input and output of the analog filter $H(s)$, while the input and output of the designed digital filter $H_D(z)$ are denoted by $x_D(k)$ and $y_D(k)$, respectively. The order of the digital filter is denoted by n_D.

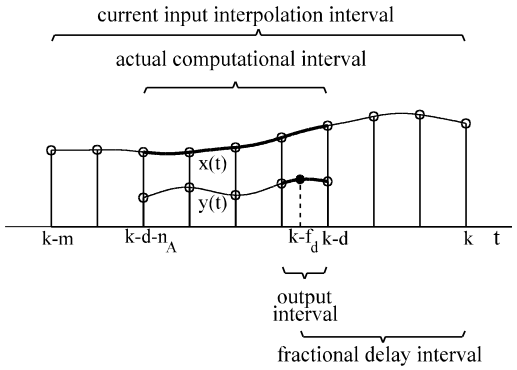

Fig. 1. Definition of the EWD methods for an analog prototype of order n_A=4 and delay d=3: the original EWD method takes m_0=0 and m_f=m=9; the R-EWD is done with m_f=n_A+d=7 and a shorter "current input interpolation interval," $k\text{-}d\text{-}n_A \le t \le k$ (the m_0 parameters are not evidenced in this figure). For each k, only the output values at t=k-d and t=k-f_d are computed.

In this section, the fixed-length recursive and nonrecursive interpolators are first defined as standalone devices. The standalone interpolator is a ***sampled-data system*** with discrete-time input and continuous-time output. Specifically, the continuous-time output signal $x_C(t)$ is built sequentially, one sampling period at a time, from the input signal $x_D(k)$. Let $\mathcal{P}$ be the interpolation space spanned by a given set of (m+1) linearly independent functions collected in the vector $\boldsymbol{\phi}(t) = [\phi_1(t),\phi_2(t),...,\phi_{m+1}(t)]$. Also, for each sampling time k, let $x_k(t)$ be the unique function that passes through (m_f+1) input samples ($m_f \le m$) placed at $t = k\text{-}m_f$, $k\text{-}m_f$ +1, ..., k and, at the same time, satisfies m_0 additional conditions, where $m_0 = m\text{-}m_f \ge 0$. Since, in general, the last few sampling periods of the interval $k\text{-}m_f \le t \le k$ exhibit relatively large extraneous oscillations [6], there are applications where it is advantageous to select the segment $k\text{-}d\text{-}1 \le t \le k\text{-}d$ of $x_k(t)$ as the "current" output segment, where d is an integer ($d < m_f/2$). The conventional (nonrecursive) interpolation is obtained when m_0= 0, whereas $m_0 \ge 1$ offers a few interpolation options, depending on the particular m_0 additional conditions.

According to the theory of sampled-data systems [4], the above interpolator is uniquely defined by the input-output equation

$$X_C(s) = X_D(z)\, H_I(s) \tag{2}$$

that relates its transfer function $H_I(s)$ to the transforms $X_D(z)$ and $X_C(s)$. In order to determine $H_I(s)$, as well as the corresponding frequency response $H_I(j\omega)$, an input signal $x_D(k)$ should be chosen followed by the sequential computation of $x_C(t)$. Finally, the transforms $X_D(z)$ and $X_C(s)$ should be computed to get

$$H_I(s) = \frac{X_C(s)}{X_D(z)}, \quad \text{and} \quad H_I(j\omega) = \left.\frac{X_C(s)}{X_D(z)}\right|_{s=j\omega,\ z=e^{j\omega}} \tag{3}$$

First of all, Eq. (3) requires a careful match of the signals $x_D(k)$ and $x_C(t)$ in conjunction with the above sequential construction that defines the standalone interpolator. Second, the expression $X_D(z)$ is usually easy to obtain whereas, depending on the basis of the interpolation space $\mathcal{P}$, the closed-form expressions of $H_I(s)$ and $H_I(j\omega)$ may be hard to derive, in spite of the fact that their final form will be shown to be relatively simple. The following two methods stand out from a practical point of view.

The impulse response method. Perform the sequential construction of the response $h_I(t)$ to an input impulse $\delta(k)$, then compute the Laplace and Fourier transforms of $h_I(t)$.While the derivation of the closed-form expressions of $H_I(s)$ and $H_I(j\omega)$ is difficult, the numeric computation of $H_I(j\omega)$ is expedited by using the FFT.

The characteristic function method. Define the *characteristic function* associated with the interpolator as the signal $x(t)$ in $\mathcal{P}$ that satisfies m_0 additional conditions and interpolates the (m_f +1)-point sequence {0,0,...,0,1}, where m_f=m-m_0, the zeros correspond to the vector $\mathbf{t}_i^T = [-m_f+d+1, ..., 0, ..., d]$ and the one is placed at $t_1 = d+1$. A few characteristic functions are presented below:

$x_0(t,d)$ – standard (nonrecursive) interpolation with $m_f = m$;

$x_1(t,d,m_0)$ – (m_f +1)-point interpolation with functions in $\mathcal{P}$, such that the first m_0 derivatives at t=0 are zero and $m_f = m - m_0$;

$x_2(t,d,m_0)$ – (m_f +1)-point interpolation with functions in $\mathcal{P}$, such that the derivatives at t = 0,1, m_0 -1 are zero and $m_f = m - m_0$.

$x_3(t,d,\mathbf{t}_d)$ – (m +1)-point ***fractional-time*** interpolation with functions in $\mathcal{P}$, such that the vector $\mathbf{t}_d$ of the zero-crossing points of $x_3(t,d,\mathbf{t}_d)$ contains the above vector $\mathbf{t}_i$ of sampling points and the vector $\mathbf{t}_r$ of m_0 time instants in-between the sampling times;

It will be shown below that the numerical implementation of the last three characteristic functions requires a recursive update of the additional m_0 parameters. Next, the one-sided z-transform $X_D(z)$ of the input sequence has a simple closed-form expression that is easily derived for m_0=0. For the general case ($m_0 \ge 1$), both one-sided transforms $X_D(z)$ and $X(s)$ in (3) above can be numerically computed to yield (3). By construction, the characteristic function $x(t)$ guarantees that ***any selected segment*** of an interpolated signal belongs to a function that passes through the last m_f input samples. Thus, both methods above yield the same expression for $H_I(s)$.

3. THE EWD AND R-EWD METHODS

The EWD methods use one of the above interpolation methods applied to the last (m_f+1) input samples $x_D(k)$ up to the current time t=k to obtain an expression $x(t)$ for the interpolated input. The current output $y_D(k)$ is computed as the analog output at the time instant t=k-d, where d is an *optional delay*. The original EWD method, exemplified in Fig. 1 takes m_0=0 and m_f=m, whereas the R-EWD corresponds to m_f=n_A+d and amounts to a computation of $x(t)$ based on the (m_f+1) points and the recursive updating of the m_0 parameters. In either case, each new output value is actually calculated from the response of the analog prototype $H(s)$ to $x(t)$ determined along the subinterval $k\text{-}d\text{-}n_A \le t \le k\text{-}d$, with ***a particular set of initial conditions***. Specifically, along this time interval, the function $y(t)$, calculated at t=k-d-n_A, . . . , k-d-1, must take on precisely the last n_A sampled values previously computed. Apparently, the EWD approach makes the approximation $y(t)$ benefit of the increased accuracy of the central part of the m-segment interpolation interval which contains most of the signal energy [6]. In addition, the R-EWD is expected to provide a better digitizing accuracy due to the smooth left end of the interpolated signal.

3.1 The recursive equations of R-EWD interpolators

In the following, the R-EWD interpolation of the input signal will be defined by

$$x(t) = \sum_{n=1}^{m+1} c_n\, \phi_n(t)\ , \qquad m = m_f + m_0\ , \tag{4}$$

where $\phi_n(t)$, n=1,2,...,m are the basis functions in $\mathcal{P}$.

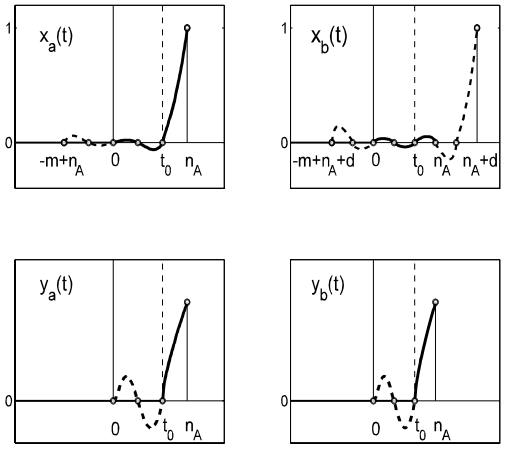

Fig. 2. I/O signals defining the design based on the characteristic functions $x(t)$: the current EWD or R-EWD output $y(n_A)$ is computed from the response of $H(s)$ to the segment $x(t)$, $0 \le t \le n_A$, of the (m_f+1)-point interpolation, with the auxiliary conditions $y(0) = y(1) = \ldots = y(n_A-1) = 0$. The EWD parameters are: (*left*) d=0, m_f=m=5, n_A=3, and m_0=0; (*right*) d=2, m_f=m=7, n_A=3, and m_0=0. The R-EWD figures are directly obtained from the figures above simply by ignoring the left-hand sides; the R-EWD parameters are: (*left*) d=0, m_f=n_A=3; (*right*) d=2, n_A=3, m_f=n_A+d=5 (the effect of the m_0 parameters is illustrated in Fig. 3).

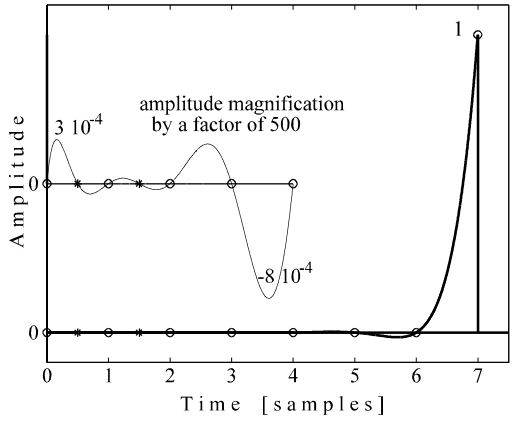

Fig. 3. A typical R-EWD characteristic function $x_3(t,d,\mathbf{t}_d)$ with parameters d=0, m_f=n_A=7, m_0=2, m=m_f+m_0=9, and $\mathbf{t}_r^T$=[0.5,1.5].

The R-EWD method is presented below for the practical case of the characteristic function $x_3(t,d,\mathbf{t}_d)$. Although the derivations for $x_1(t,d,m_0)$ and $x_2(t,d,m_0)$ follow closely this derivation, they are only of academic interest due to the numerical problems involving signal derivatives. At the same time, the original EWD method corresponding to $x_0(t,d)$ has been already presented in [1] and [7]. Now, considering the interpolator that corresponds to the characteristic function $x_3(t,d,\mathbf{t}_d)$, m is selected such that the coefficients c_n are obtained as the *exact solution* of the (m+1) algebraic equations expressed in vector form

$$\begin{bmatrix} \mathbf{A}_i \\ \mathbf{A}_r \end{bmatrix} \mathbf{c} = \begin{bmatrix} \mathbf{x}_i \\ \mathbf{x}_r \end{bmatrix}, \tag{5}$$

where $\mathrm{A}_i = \boldsymbol{\phi}(\mathbf{t}_i)$, $\mathrm{A}_r = \boldsymbol{\phi}(\mathbf{t}_r)$, $\boldsymbol{\phi}(t) = [\phi_1(t), \phi_2(t), \ldots, \phi_{m+1}(t)]$, $\mathbf{t}_i$ is the column-vector of the last (m_f+1) sampling points, and $\mathbf{t}_r$ is the column-vector of the m_0 time instants in-between the sampling times; also, $\mathbf{x}_i$ is the known vector of the last (m_f+1) input samples, while $\mathbf{x}_r$ is the "unknown" vector of the input values at the time instants in $\mathbf{t}_r$. For each new current time $(k+1)$, $\mathbf{x}_r$ is updated with the interpolated values $x(\mathbf{t}_r+1)$ calculated at the previous time k. Thus, defining the new matrices B and V with the relationship

$$[\mathbf{B}, \mathbf{V}] = \begin{bmatrix} \mathbf{A}_i \\ \mathbf{A}_r \end{bmatrix}^{-1}, \tag{6}$$

the following recursive equations are obtained:

$$\mathbf{c}(k) = \mathrm{B}\, \mathbf{x}_i(k) + \mathrm{V}\, \mathbf{x}_r(k), \tag{7}$$

$$\mathbf{x}_r(k+1) = \boldsymbol{\Phi}(\mathbf{t}_r + 1)\, \mathbf{c}(k), \tag{8}$$

or

$$\mathbf{x}_r(k+1) = \mathrm{G}\, \mathbf{x}_i(k) + \mathrm{U}\, \mathbf{x}_r(k), \tag{9}$$

where $\mathrm{G} = \boldsymbol{\phi}(\mathbf{t}_r+1)\,\mathrm{B}$ and $\mathrm{U} = \boldsymbol{\phi}(\mathbf{t}_r+1)\,\mathrm{V}$. Finally, (9) may be viewed as a state equation with the state variable $\mathbf{x}_r$.

Therefore, an arbitrary initial vector $\mathbf{x}_{r0}$ may always start the interpolation process provided that the eigenvalues λ_n, n=1,..., m_0, of the m_0x m_0 matrix U are strictly within the unit circle. Moreover, the updating process can be considered terminated after a number K of sampling periods that make all the values λ_n^K negligible. As the practice shows, these conditions can be always met, and so a trial and error step of the design is to be performed in order to determine the largest admissible m_0 and $\mathbf{t}_r$ that will provide the smoothest possible interpolation.

3.2 The Time-Domain Invariance Synthesis

The principle of time-domain invariance used to digitize analog systems assumes that identical inputs, $x_D(k)=x(t)|_{t=k}$, yield identical outputs, $y_D(k)=y(t)|_{t=k}$, along the entire time axis. Thus, this time-domain equivalence [5] implies the identity of the z-transforms $Y_D(z) \triangleq H_D(z)X_D(z)$ and $Z\{Y(s)\}$. With the notation of [4, Sect. 4.3.1], the expression $Z\{Y(s)\} \triangleq Z\{[\mathcal{L}^{-1}[Y(s)]]|_{t=k}\}$ is the z-transform of the samples of $y(t)$. Thus, the above identity yields the transfer function

$$H_D(z) = \frac{1}{X_D(z)}\, Z\{Y(s)\}\,. \tag{10}$$

The crux of the design is the selection of the characteristic function $x(t)$, together with the method for calculating the corresponding response $y(t)$ of the analog prototype $H(s)$. Following the same line of the analysis performed in [7], the class of *admissible characteristic functions*, $x(t)$, corresponding to an (m_f+1)-point interpolation will be the set of functions whose zeros are m_f adjacent integers, and satisfy m_0 additional conditions ($m_0 \ge 0$). Figs. 2 and 3 illustrate the general shape of the admissible characteristic functions $x(t)$, as well as the corresponding form of the response $y(t)$ of the analog filter. Moreover, Fig. 2 shows that the time origin is chosen such that there are n_A interpolation segments between t=0 and the time instant where the current digital output is computed.

3.3 Derivation of R-EWD transfer functions

The characteristic function $x_3(t,d,\mathbf{t}_d)$ is now considered and m is chosen such that the coefficients c_n in (4) are obtained as the *exact solution* of a modified set (5) of (m+1) algebraic equations.

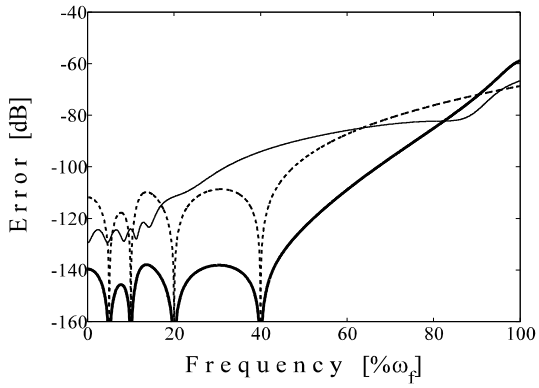

Fig.4. The digitizing error $E_D(\omega) = 20\log_{10}(|e^{-jd\omega}H(j\omega)-H_D(e^{j\omega})|)$ for three digital equivalents of order n_D=7 of the transfer function $H(s)=32(s^2+0.2s+2)/[(8s^2+4s+1)(4s^2+s+1)(s+1)]$. *Thin solid line*: the iterative WLS design. *Dashed line*: the EWD filter. *Thick solid line:* the R-EWD filter (sampling period = 0.75 sec, d=0, m_f=n_A=5, m_0=2, m=m_f+m_0=7, and $\mathbf{t}_r^T$=[1/3,2/3]).

This time, the matrices $A_i = \boldsymbol{\phi}(\mathbf{t}_i)$ and $A_r = \boldsymbol{\phi}(\mathbf{t}_r)$ are computed for the vectors $\mathbf{t}_i^T$=[0,1,...,m_f] and $\mathbf{t}_r$ containing the m_0 fractional times within the interval [0, m_f]; $\mathbf{x}_i^T$=[0,...,0,1] is a vector with (m_f+1) components and $\mathbf{x}_r$ is a vector containing m_0 zeros. For computational reasons, the functions $\phi_n(t)$ are selected as linearly independent functions with simple one-sided Laplace transforms. The new form of (5) will be referred to in the following as (5'). Accordingly, the one-sided Laplace and z-transforms of $x(t)$ and its sampled form $x_D(k)$, are rational functions of the form

$$X(s) = \frac{R(s)}{Q(s)}, \qquad X_D(z) = \frac{z\,P_{m_0}(z)}{Q_D(z)}, \tag{11}$$

where both denominators are monic polynomials with the same degree (m+1). The derivation of $R(s)$ from (4) and (5') is tedious but straightforward. Next, if m_0=0, the numerator of $X_D(z)$ is simply z. This is an immediate result of the fact that all the values of the characteristic function at the sampling times $-\infty < t \le n_A+d-1$ are zero, while the first nonzero value $x(n_A+d)=1$ yields the relationship $\lim_{z\to\infty} z^{nA+d} X_D(z) = 1$. If $m_0 \ge 1$, the relations (3), (7), and (9) can be used to show that $P_{mo}(z)$ becomes a monic polynomial with the roots precisely the eigenvalues λ_n of the matrix U in (9). At the same time, the polynomials $Q(s)$ and $Q_D(z)$ are defined by their roots. Indeed, the roots of $Q(s)$ are the poles r_n of the Laplace transforms of $\phi_n(t)$, whereas the roots of $Q_D(z)$ are $z_n = e^{r_n}$. Finally, in order to satisfy the initial conditions $y(t)=0$, for $t = 0, 1, \ldots, n_A-1$, as illustrated in Fig. 2, the expression of $Y(s)$ in (10) exhibits both a zero-state and a zero-input component: $Y(s)=Y_{zs}(s)+Y_{zi}(s) = X(s)H(s)+Y_{zi}(s)$. A first result is

$$\begin{aligned} Z\{X(s)H(s)\} &= Z\left\{\frac{R(s)\,H(s)}{Q(s)}\right\} \\ &\triangleq Z\left\{\frac{b_0 s^{n_A+m-1} + \ldots + b_{n_A+m-1}}{Q(s)\,(s^{n_A} + a_1 s^{n_A-1} + \ldots + a_{n_A})}\right\} \\ &\triangleq \frac{z\,(\beta_0 z^{n_A+m} + \ldots + \beta_{n_A+m})}{Q_D(z)\,(z^{n_A} + g_1 z^{n_A-1} + \ldots + g_{n_A})}, \end{aligned} \tag{12}$$

whith the polynomial $z^{nA} + g_1 z^{nA-1} + \ldots + g_{nA}$ defined by its roots $z_n = e^{s_n}$, where s_n, n=1,...,n_A, are the poles of $H(s)$. Then,

$$H_D(z) = \frac{Q_D(z)}{z\,P_{m_0}(z)}\left[\frac{z(\beta_0 z^{n_A+m} + \ldots + \beta_{n_A+m})}{Q_D(z)\,(z^{n_A} + g_1 z^{n_A-1} + \ldots + g_{n_A})} + \frac{z\,(f_1 z^{n_A-1} + \ldots + f_{n_A})}{z^{n_A} + g_1 z^{n_A-1} + \ldots + g_{n_A}}\right]. \tag{13}$$

In order to satisfy the initial conditions, the coefficients f_k must be chosen such that the n_A leading terms of the numerator of $H_D(z)$ in (13) are canceled. This is achieved with the recursive equations

$$Q_D(z) \triangleq z^{m+1} + d_1 z^m + \ldots + d_{m+1}, \quad f_1 = -\beta_0, \quad \left\{ f_k = -\beta_{k-1} - \sum_{n=1}^{k-1} f_{k-n}\, d_n, \quad k = 2, \ldots, n_A \right\}. \tag{14}$$

4. CONCLUSIONS

The main contribution of the paper is the incorporation of the new class of recursive interpolators into the previous EWD method [1] that was developed for the computation of digital equivalents of analog filters with application to fractional delay and sampling rate conversion [2], [7]. The smoothening properties of the recursive interpolators support the claim that the proposed R-EWD filters provide lower digitizing errors than the original EWD filters. The latter were proven in [7] to be near-optimal equivalents of their analog counterparts. Thus, the proposed filters are not only efficient tools for FD applications, but also very accurate digital equivalents of the prototype filters (designed in analog form) whose delays are to be controlled. Some of these considerations are illustrated in Fig. 4. It is worth noting that the poles of the R-EWD and WLS filters (including the m_0 extraneous poles) are remarkably close for reasons related to the analyticity of $H(s)$ and $H_D(z)$ [7]. Yet, the accuracy of the R-EWD filter is about two orders of magnitude better than that one of the other two filters.

5. REFERENCES

[1] G.I. Braileanu, "Extended-Window Interpolation Applied to Digital Filter Design," IEEE Trans. Signal Processing, vol. 44, pp. 457-472, 1996.

[2] G.I. Braileanu, "Digital Filters with Implicit Interpolated Output," IEEE Trans. Signal Processing, vol 45, pp. 2551-2560, 1997.

[3] A. V. Oppenheim, A. S. Willsky, and I. T. Young, "Signals and Systems," Englewood Cliffs, NJ: Prentice Hall, 1983.

[4] G. F. Franklin, J. D. Powell, and M. L. Workman, "Digital Control of Dynamic Systems," Menlo Park, CA: Addison Wesley, 1998.

[5] R. E. Ziemer, W. H. Tranter, and D. R. Fannin, "Signals and Systems: Continuous and Discrete," Upper Saddle River, NJ: Prentice Hall, 1998.

[6] M.J.D. Powell, "Approximation Theory and Methods," Cambridge Univ. Press, 1981.

[7] G. Braileanu, "Equivalence Between the Extended Window Design of IIR Filters and Least Squares Frequency Domain Designs," Proc. 2003 Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP'03), vol. VI, pp. 21-24, Hong Kong, 2003.