Design Methodology for Nearly Linear-Phase Recursive Digital Filters by Constrained Optimization

David Guindon, Dale J. Shpak, *Senior Member, IEEE*, and Andreas Antoniou, *Life Fellow, IEEE*

***Abstract*—A methodology for the design of recursive digital filters having nearly linear phase response is proposed. The underlying design method is of the direct type whereby the filter is designed as a single unit. The design problem is formulated as a cascade of filter sections where each section is represented by a biquadratic transfer function either in the conventional polynomial form or in the polar form. The design problem is then solved using a constrained Newton's method whereby constraints are used to assure the stability of the filter, to control the step size in order to achieve fast convergence, and to eliminate a real-axis pole-migration problem that often interferes with the design process. Several design examples demonstrate that when compared with filters designed using existing state-of-the-art methods, the proposed methodology yields filters having reduced order and/or improved performance.**

***Index Terms*—Recursive digital filters, infinite-impulse response (IIR) digital filters, nearly linear-phase filters, constant-delay filters.**

I. Introduction

RECURSIVE (IIR) digital filters are often used because high selectivity can be achieved with a significantly lower filter order than that required by nonrecursive (FIR) filters satisfying the same specifications. However, high selectivity is accompanied by an often undesirable nonlinear phase response which corresponds to a varying group delay with respect to frequency. A nonlinear phase response would thus result in temporal skewing of the various frequency components of the signal. Such signal distortion, usually referred to as *phase* or *delay distortion* [1] can cause audible artifacts in audio signals, difficulties for symbol detection in communication systems, or skewed and blurred features in images.

In some applications, phase response is unimportant. For such applications, a fairly large choice of methods is available to the filter designer ranging from *closed-form* methods based on the classical analog-filter approximations (see [1, Ch. 12]) to optimization methods of all kinds. In [1, Ch. 16], unconstrained algorithms of the quasi-Newton family are used in a least-pth formulation. Filter stability is achieved by means of a well-known stabilization technique whereby poles outside the unit-circle of the z plane are replaced by their reciprocals. MATLAB® function *iirlpnorm* in [2] implements an unconstrained least-pth quasi-Newton algorithm of the type found in [1]. On the other hand, MATLAB® function *iirlpnormc* in [2] implements a least-pth Newton method that uses barrier constraints to assure the stability of the resulting filter. A more recent method reported in [3] uses a constraint transcription method to achieve an efficient solution of a non-smooth non-convex objective function with continuous functional constraints. Since there is no restriction on the resulting phase response, these methods generally yield a better magnitude response than methods that also optimize the phase response.

In other applications, it is important to have linear or nearly linear phase response in addition to a prescribed magnitude response. If an absolutely linear phase response over the entire baseband is, for some reason, required, then a nonrecursive design must be used. However, in most practical applications a nearly linear phase response with respect to passbands is more than adequate considering that signals in stopbands are generally deemed to be noise, and introducing delay distortion in noise is of no consequence. The classical method for designing recursive filters with nearly linear phase response with respect to passbands is the *equalizer approach* whereby a recursive filter that would satisfy the magnitude response specifications is first designed by some means and then a delay equalizer in the form of an allpass filter is designed to compensate for the fluctuations in the group delay of the recursive filter with respect to passbands [1], [4], [5]. An approach for the design of equalizers using a constrained least-pth primal active-set method is described in [6] and a MATLAB® function *iirgrpdelay* which implements a constrained Newton method that can be used to design arbitrary delay equalizers can be found in [7].

A great variety of other methods have also been used for the design of recursive filters having nearly linear phase response. In a method described in [8], filter stability is assured by an apparently heuristic method whereby the step size is halved during each iteration. In [9], a Gauss–Newton method is used in conjunction with a multiple exchange method and Rouchés theorem to assure the stability of filters optimized to meet prescribed magnitude and phase responses. The weighted integral of the squared-error method in [10] allows the inclusion of stability constraints with the magnitude constraints. Another method reported in [11] uses a convex stability domain so that the problem can be solved using semidefinite programming. In [12] a so-called *p-power* error criterion is proposed and is then optimized using quadratic programming; this approach is then applied for the design of one- and two-dimensional digital filters. In [13], a globally-optimal solution for a recursive filter having a prescribed group delay is found using polynomial optimization. The constrained optimization problem is simplified in [14] by using an argument-based stability criterion that

Manuscript received November 04, 2008; revised April 27, 2009; accepted October 01, 2009. Date of publication December 28, 2009; date of current version July 16, 2010. This work was supported by Natural Sciences and Engineering Research Council of Canada. This paper was recommended by Associate Editor E. A. B. da Silva.

The authors are with the Department of Electrical and Computer Engineering, University of Victoria, Victoria, BC V8W 3P6, Canada (e-mail: daveguindon@gmail.com; dshpak@ece.uvic.ca; aantoniou@ieee.org).

Digital Object Identifier 10.1109/TCSI.2009.2035412

1549-8328/$26.00 © 2010 IEEE

results in a single integer-valued constraint. In [15], a design technique of the model-reduction family is described whereby second-order cone programming is used to minimize an appropriate objective function. With this technique, IIR filters that satisfy prescribed passband and stopband magnitude-response specifications can be designed.

In this paper, a new methodology for the design of recursive filters with nearly linear phase response is proposed. The underlying method is of the *direct type* whereby the filter is designed as a single unit in such a way as to achieve prescribed magnitude-response specifications on the one hand and nearly flat group-delay characteristic with respect to passbands on the other while keeping the filter order as low as possible. The problem is formulated as a constrained quadratic programming problem and is solved using an optimization engine based on Newton's method. Two different forms of the transfer function, namely, a *polynomial* and a *polar* form, are explored and the two are then compared. The paper deals with several practical issues associated with the design of recursive digital filters in general such as initialization and the handling of stability and step-size constraints to achieve stability on the one hand and to achieve fast convergence on the other. The paper also deals with a *real-axis pole-migration problem* that often interferes with the design process. Numerous real-life designs have shown that the proposed methodology leads to a significant reduction in the filter order required to satisfy arbitrary specifications imposed on the magnitude-response and group delay relative to corresponding designs obtained by some of the existing state-of-the art methods including the classical equalizer approach.

The paper is organized as follows. Section II deals with the problem formulation and Sections III and IV deal with the polynomial and polar formulations, respectively. In Section V, practical issues pertaining to initialization and termination criteria are investigated. The polynomial and polar formulations are compared in Section VI. Typical design examples and results are reported in Section VII and conclusions are drawn in Section VIII. Appendixes A – C provide closed-form formulas for the gradient vector, Hessian matrix, and for the required constraints.

II. Problem Formulation

Recursive digital filters are often realized in terms of a cascade arrangement of second-order biquadratic sections in order to achieve reduced sensitivity to roundoff errors. The transfer function of an n-order filter of this type can be represented in the classical *polynomial* form [1]

$$H(\mathbf{x}_1, z) = H_0 \prod_{m=1}^{J} \frac{a_{0m} + a_{1m}z + z^2}{b_{0m} + b_{1m}z + z^2} = H_0 \prod_{m=1}^{J} \frac{N_m(z)}{D_m(z)} \tag{1}$$

where

$$\mathbf{x}_1 = [a_{01}\, a_{11}\, b_{01}\, b_{11} \cdots b_{0J}\, b_{1J}\, H_0]^T \tag{2}$$

J is the number of filter sections, $n = 2J$ is the filter order, and H_0 is a positive multiplier constant. An odd-order transfer function can be obtained by including a first-order section. The transfer-function coefficients are assumed to be real.

Following the approach described in [16], the transfer function can be expressed in the so-called *polar form* as

$$\begin{aligned} H(\mathbf{x}_2, z) &= H_0 \prod_{m=1}^{J} \frac{(z - r_{am}e^{j\theta_{am}})(z - r_{am}e^{-j\theta_{am}})}{(z - r_{bm}e^{j\theta_{bm}})(z - r_{bm}e^{-j\theta_{bm}})} \\ &= H_0 \prod_{m=1}^{J} \frac{r_{am}^2 - 2r_{am}\cos(\theta_{am})z + z^2}{r_{bm}^2 - 2r_{bm}\cos(\theta_{bm})z + z^2} \end{aligned} \tag{3}$$

where the r_{am} and r_{bm} are the zero and pole radii and θ_{am} and θ_{bm} are the corresponding angles in the z plane, respectively, and

$$\mathbf{x_2} = [r_{a1}\, \theta_{a1}\, r_{b1}\, \theta_{b1} \cdots r_{bJ}\, \theta_{bJ}\, H_0]^T. \tag{4}$$

Although (1) and (3) are equivalent, there are, nevertheless, significant differences when performing the optimization, notably in the evaluation of the constraint matrices. Consequently, the performance of the optimization algorithm as well as that of the resulting digital filters can be affected as will be discussed in Section VI. The two formulations are developed in Sections III and IV below.

The filter design problem is formulated as a quadratic programming problem of the form

$$\begin{aligned} &\underset{\mathbf{x}}{\text{minimize}}\, E(\mathbf{x}) \\ &\text{subject to } \mathbf{A}^T\mathbf{x} \le \mathbf{b} \end{aligned} \tag{5}$$

where $\mathbf{x}$ is the vector of optimization parameters and matrix $\mathbf{A}$ and vector $\mathbf{b}$ represent linear constraints. This problem is solved by using a constrained Newton's method [17], [18]. In addition to $\mathbf{x}$, $\mathbf{A}$, and $\mathbf{b}$, Newton's method requires the objective function $E(\mathbf{x})$, the gradient vector $\mathbf{g}(\mathbf{x})$, and the Hessian matrix $\mathbf{H}(\mathbf{x})$; the last two items will be referred to simply as the *gradient* and *Hessian*, respectively, hereafter.

The optimization algorithm minimizes the approximation error $E(\mathbf{x})$ over a set of frequencies $\mathbf{\Omega} = \{\omega_i, 1 \le i \le M\}$ where M is sufficiently large to ensure that the discretized problem is a good approximation of the corresponding continuous problem. Frequencies ω_i can be chosen by using the nonuniform adaptive sampling technique proposed in [19]. This technique is described in some detail in Section VII.

The approximation error e_i at frequency $\omega_i \in \mathbf{\Omega}$ is defined as

$$e_i(\mathbf{x}) = F(\mathbf{x}, \omega_i) - F_0(\omega_i) \tag{6}$$

where $F(\mathbf{x}, \omega_i)$ the actual frequency response, i.e., the transfer function evaluated at $z = e^{j\omega_i T}$, $F_0(\omega_i)$ is the desired frequency response, and T is the sampling period in seconds, typically assumed to be unity.

The objective function $E(\mathbf{x})$ can be formulated as a weighted L_p function of the form

$$E(\mathbf{x}) = \sum_{i=1}^{M} w_i |e_i(\mathbf{x})|^p \tag{7}$$

where $p > 0$ is an even integer and $\{w_i, 1 \le i \le M\}$ is a set of error weights corresponding to the frequencies ω_i in $\mathbf{\Omega}$. Suitable values for the weights in order to achieve prescribed specifications can be assigned by using the weighting technique described on p. 747 of [1].

Often a value of $p = 2$ is used which would result in least-squares solutions. However, minimax solutions can also be obtained by incorporating the above formulation in a least-pth algorithm such as the one described in [20] (see also [1, p. 739]).

A. Gradient of the Objective Function

The gradient $\mathbf{g}(\mathbf{x})$ of the objective function in (7) can be evaluated as

$$\begin{aligned}\nabla E(\mathbf{x}) &= \frac{p}{2}\sum_{i=1}^{M} w_i |e_i(\mathbf{x})|^{p-2}[\bar{e}_i(\mathbf{x})\nabla e_i(\mathbf{x}) + e_i(\mathbf{x})\nabla \bar{e}_i(\mathbf{x})] \\ &= p\sum_{i=1}^{M} w_i |e_i(\mathbf{x})|^{p-2}\mathrm{Re}[\bar{e}_i(\mathbf{x})\nabla e_i(\mathbf{x})] \qquad (8)\end{aligned}$$

where $\bar{e}_i(\mathbf{x})$ is the complex conjugate of $e_i(\mathbf{x})$ and $\nabla e_i(\mathbf{x})$ is the gradient of the error function with respect to $\mathbf{x}$.

B. Hessian of the Objective Function

The Hessian $\mathbf{H}(\mathbf{x})$ of the objective function in (7) is

$$\nabla^2 E(\mathbf{x}) = p\sum_{i=1}^{M} w_i \mathbf{H}_i^{(E_x)}(\mathbf{x}) \qquad (9)$$

where

$$\begin{aligned}\mathbf{H}_i^{(E_x)}(\mathbf{x}) = &(p-2)|e_i(\mathbf{x})|^{p-4} \\ &\cdot \mathrm{Re}[\bar{e}_i(\mathbf{x})\nabla e_i(\mathbf{x})]\mathrm{Re}[\bar{e}_i(\mathbf{x})\nabla^T e_i(\mathbf{x})] \\ &+ p|e_i(\mathbf{x})|^{p-2}\mathrm{Re}[\bar{e}_i(\mathbf{x})\nabla^2 e_i(\mathbf{x}) \\ &+ \nabla\bar{e}_i(\mathbf{x})\nabla^T e_i(\mathbf{x})] \qquad (10)\end{aligned}$$

and $\nabla^2 e_i(\mathbf{x})$ represents the Hessian of the error function.

C. Positive-Definiteness of the Hessian

For Newton's method to converge, the Hessian must always be positive definite. However, during the optimization process this condition can be violated since we are dealing with a highly nonconvex multimodal problem in the present paper. This difficulty can be circumvented by modifying the Hessian, $\mathbf{H}$, in each iteration such that

$$\mathbf{H}^* = \mathbf{H} + \gamma \mathbf{I}_n$$

where $\mathbf{I}_n$ is the $n \times n$ identity matrix and

$$\begin{aligned}\gamma > -\lambda_n, &\quad \text{if } \lambda_n \le 0 \\ \gamma = 0, &\quad \text{if } \lambda_n > 0\end{aligned}$$

with λ_n being the minimum eigenvalue of $\mathbf{H}$ [17]. In this way, a positive definite Hessian is achieved in every iteration which would assure that the Newton direction is always a descent direction.

III. Polynomial Formulation

For a linear phase response, the desired frequency response assumes the form

$$F_0(\omega_i) = M_0(\omega_i)e^{-j\omega_i\tau} \qquad (11)$$

where $M_0(\omega_i)$ is the desired magnitude response at ω_i and τ is the group delay. Usually, the absolute value of the group delay is unimportant and in such applications the appropriate course of action is to treat the group delay as an independent variable that can be chosen to bring about an additional reduction in the approximation error [5].

The actual frequency response can be obtained from (1) as

$$\begin{aligned}F(\mathbf{x}_1, \omega_i) &= H_0 \prod_{m=1}^{J} \frac{a_{0m} + a_{1m}e^{j\omega_i T} + e^{j2\omega_i T}}{b_{0m} + b_{1m}e^{j\omega_i T} + e^{j2\omega_i T}} \\ &= H_0 \prod_{m=1}^{J} \frac{N_m(e^{j\omega_i T})}{D_m(e^{j\omega_i T})} \qquad (12)\end{aligned}$$

and if the group delay, τ, is treated as another independent variable, the parameter vector in (2) assumes the form

$$\mathbf{x}_1 = [a_{01}\, a_{11}\, b_{01}\, b_{11} \cdots b_{0J}\, b_{1J}\, H_0\, \tau]^T. \qquad (13)$$

The approximation error, $e_i(\mathbf{x})$, given by (6), can be computed by using (11) and (12). The elements of the gradient of the error function $\nabla e_i(\mathbf{x})$ and the Hessian of the error function $\nabla^2 e_i(\mathbf{x})$ required in (8) and (10), respectively, are given in Appendix A.

A. Constraints

The linear constraint matrix $\mathbf{A} = [\mathbf{A}_1\ \mathbf{A}_2\ \mathbf{A}_3]^T$ consists of three submatrices corresponding to step-size, filter-stability, and real-pole boundary constraints, respectively. Similarly, the constraint vector can be expressed as $\mathbf{b} = [\mathbf{b}_1\ \mathbf{b}_2\ \mathbf{b}_3]^T$. Closed-form formulas for the elements of matrix $\mathbf{A}$ and vector $\mathbf{b}$ can be found in Appendix B.

1) Step-Size Constraints: These constraints limit the changes in the independent parameters to prevent the algorithm from overshooting the minimum point, which could cause the algorithm to diverge. It turns out that, in practice, the sensitivities of the objective function to the various elements of the parameter vector $\mathbf{x}$ in (13) tend to differ quite a bit and by applying different step limits for the filter coefficients, the multiplier constant H_0, and the group delay τ, better results can be achieved.

In our design algorithm, step limits (upper bounds) for the different parameters are applied by using a step-size vector $\boldsymbol{\beta}^{(k)}$ of the form

$$\boldsymbol{\beta}^{(k)} = \left[\beta_c^{(k)}\, \beta_{H_0}^{(k)}\, \beta_\tau^{(k)}\right]^T \qquad (14)$$

where k is the iteration number. The different step limits are computed recursively as

$$\beta_c^{(k)} = \beta_c^{(k-1)}(1 - R_\beta), \quad \text{if } f^{(k)} > f^{(k-1)} \qquad (15)$$

$$\beta_{H_0}^{(k)} = \beta_{H_0}^{(k-1)}(1 - R_\beta), \quad \text{if } f^{(k)} > f^{(k-1)} \qquad (16)$$

$$\beta_\tau^{(k)} = \beta_\tau^{(k-1)}\left(1 - \frac{1}{2}R_\beta\right), \quad \text{if } f^{(k)} > f^{(k-1)} \qquad (17)$$

where R_β is a constant and $f^{(k)}$ is the objective function evaluated at $\mathbf{x}^{(k)}$. The step-size vector is initialized at the start of the algorithm using empirical values. On the basis of many filter designs, a good value for R_β was found to be 0.5. Whenever the value of the objective function increases, the step limits for the coefficient and transfer function constant parameters are restricted to $(1 - R_\beta)$ of their previous values and the step limit for the delay parameter is restricted to $(1 - R_\beta/2)$ of its previous value.

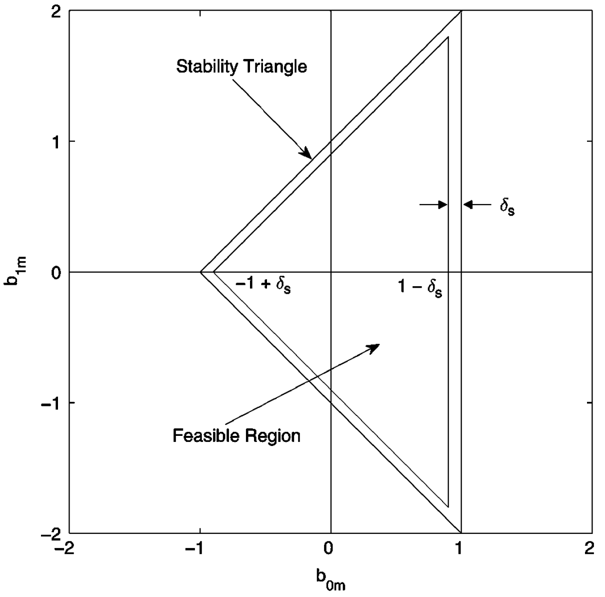

Fig. 1. Stability region with a stability margin δ_S.

2) Filter-Stability Constraints: For a stable recursive filter, the poles must lie within the unit circle of the z plane. For each biquadratic transfer function in (1), the coefficient pairs $\{b_{0m}, b_{1_m}\}$ must be located in the stability triangle shown in Fig. 1 [1]. This figure also shows an inner triangle which provides a stability margin $\delta_S > 0$ [21] which would ensure that the filter does not become unstable in the presence of coefficient roundoff errors. A small positive value of δ_S also helps to reduce the numerical sensitivity of the optimization algorithm.

Incorporating a stability margin, the stability constraints assume the form

$$b_{0m} \leq 1 - \delta_S \tag{18}$$

$$b_{1m} - b_{0m} \leq 1 - \delta_S \tag{19}$$

$$-b_{1m} - b_{0m} \leq 1 - \delta_S. \tag{20}$$

3) Real-Pole Boundary Constraints: The authors have observed that during the optimization process, a complex-conjugate pole pair may migrate to the real axis and thus become a pair of distinct real poles. Additionally,

a) such real poles often have a small radius and, therefore, do not contribute significantly to the frequency response of the filter;
b) although a conjugate pole pair will readily become a real pole pair, it is less likely that a real pole pair will become a conjugate pair.

In effect, it is beneficial to prevent pole migration to the real axis from occurring especially on account of observation (a). This is accomplished by restricting the feasibility region further. For a biquadratic filter section, the poles are real if

$$b_{0m} < \frac{1}{4}b_{1i}^2. \tag{21}$$

Applying this constraint results in the parabolic boundary shown in Fig. 2. Unfortunately, (21) is not a linear constraint and so it cannot be used as a constraint in (5). However, the parabolic boundary can be approximated by a set of tangent lines that can serve as linear constraints. Seven tangent lines located at

$$b_{1m} = \{-1.5, -1, -0.5, 0, 0.5, 1, 1.5\}$$

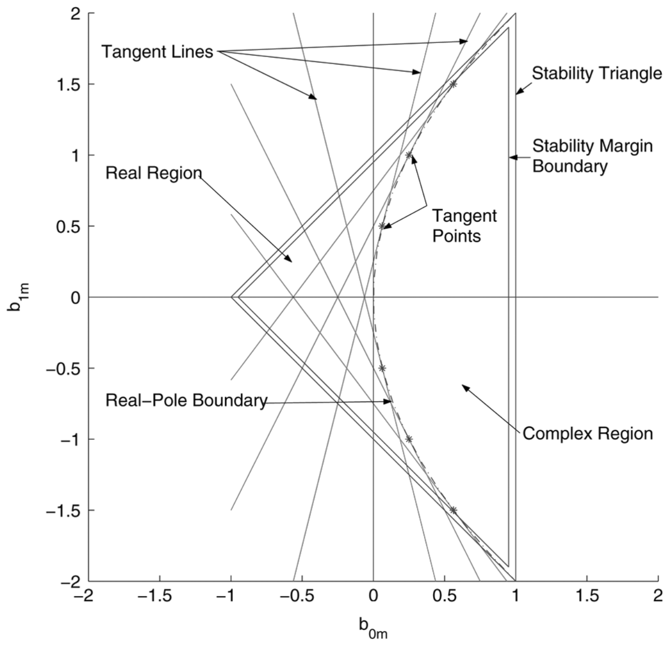

Fig. 2. The feasible region and real-pole boundary.

as illustrated in Fig. 2 were found to work well. The equations for the tangent lines are given by

$$b_{1m} = \frac{4}{3}b_{0m} + \frac{3}{4} \tag{22}$$

$$b_{1m} = 2b_{0m} + \frac{1}{2} \tag{23}$$

$$b_{1m} = 4b_{0m} + \frac{1}{4} \tag{24}$$

$$b_{0m} = 0 \tag{25}$$

$$b_{1m} = -\frac{4}{3}b_{0m} - \frac{3}{4} \tag{26}$$

$$b_{1m} = -2b_{0m} - \frac{1}{2} \tag{27}$$

$$b_{1m} = -4b_{0m} - \frac{1}{4} \tag{28}$$

and since the coefficients change from one iteration to the next one, these equations must be updated in every iteration.

IV. Polar Formulation

The actual frequency response for the polar formulation can be obtained from (3) and the error function in (6) can be deduced using (11) and (12).

With the inclusion of the group delay, τ, as an independent parameter, the parameter vector in (4) assumes the form

$$\mathbf{x} = [r_{a0}\,\theta_{a0}\,r_{b0}\,\theta_{b0}\,\cdots\,r_{bJ}\,\theta_{bJ}\,H_0\,\tau]^T. \tag{29}$$

Formulas for the elements of the gradient of the error function $\nabla e_i(\mathbf{x})$ and the Hessian of the error function $\nabla^2 e_i(\mathbf{x})$ required in (8) and (10), respectively, are given in Appendix C.

A. Constraints

As for the case of the polynomial formulation, we apply step-size, filter-stability, and real-pole boundary constraints.

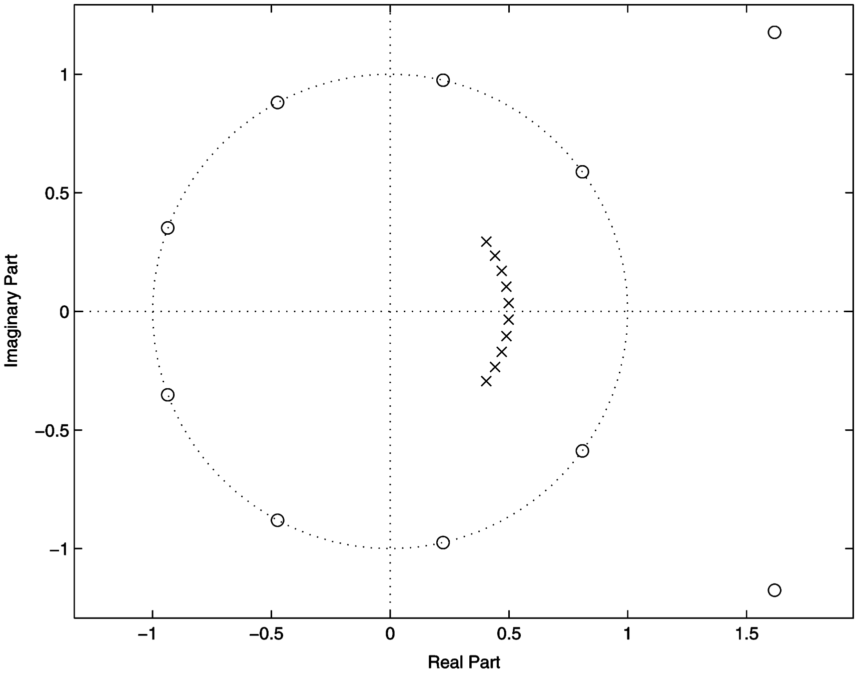

Fig. 3. Initial pole/zero placement for a 10th-order lowpass filter.

1) Step-Size Constraints: The form of these constraints is very similar to those for the polynomial formulation, i.e.,

$$\boldsymbol{\beta}^{(k)} = \left[\beta_{\hat{p}_i}^{(k)}\,\beta_{H_0}^{(k)}\,\beta_{\tau}^{(k)}\right]^T \tag{30}$$

where $\beta_{\hat{p}_i}^{(k)}$ represents the step limits for the zero/pole radii and angles for the biquadratic factors for $i = 1, \ldots, J$ for the kth iteration. The β values are initialized and modified in the same manner as for the polynomial formulation.

2) Filter-Stability Constraints: The polar formulation simplifies the stability constraints into a single constraint per section of the form $0 \leq r_{bm} < 1 - \delta_S$ where δ_S is the stability margin. The lower bound is only necessary when constraining poles to lie in the passband region as is described in the next paragraph.

3) Real-Pole Boundary Constraints: Like the stability constraints, the real-pole boundary constraints reduce to a single constraint per section of the form $0 < \theta_{bm} < \pi$. Another use of a constraint of this type is to force the poles to remain in the passband sector. In the case of lowpass filters, the constraint would assume the form $0 < \theta_{bm} < \omega_p$ where ω_p is the passband edge frequency.

It was observed through experimentation that the polar formulation is significantly less susceptible to the real-axis pole migration problem than the polynomial formulation. Consequently, the real-pole boundary constraint can often be omitted in practice in which case the constraint matrix would assume the form $\mathbf{A} = [\mathbf{A}_1\ \mathbf{A}_2]^T$.

V. Algorithm Initialization and Termination

As stated earlier, nearly linear-phase recursive filters, like phase equalizers, are highly nonconvex multimodal problems and, therefore, it is important to find good initial values for parameters a_i, b_i, H_0, and τ. For the design problem under consideration, two techniques were found to yield good initial designs.

A. Method of Trends

In [22], a large number of digital filters were designed and the successful designs were analyzed to find patterns or trends in the pole and zero locations. Also in [22], the concepts of *phase zeros* and *magnitude zeros* were introduced. As shown in Fig. 3 for a tenth-order lowpass filter, two phase zeros are located outside the unit circle and contribute to the phase response in the passband and eight magnitude zeros are located very close or on the unit circle outside the passband sector thereby resulting in increased attenuation in the stopband. On the basis of these patterns, good initial designs can be readily generated.

B. Balanced Model Truncation Method

The balanced model truncation (or reduction) method (BMT) is often used to generate a low-order recursive filter from a high-order nonrecursive filter [23]–[26]. The main steps for this technique are to first convert the transfer function of the nonrecursive filter into a state-space balanced model, then reduce the system order, and finally convert the model into the reduced-order recursive transfer function. The coefficients or zeros and poles of the resulting transfer function can then be used as the initial values of the transfer function in (1) or (3) as appropriate.

C. Termination

For some designs, the optimization algorithm would not converge to a solution that satisfied the prescribed specifications in a reasonable number of iterations. To circumvent this problem, a fail-safe upper limit of, say, 200 iterations, can be imposed on the algorithm. For designs that converge, the algorithm is terminated earlier using the progress ratio

$$\epsilon = \frac{|E_k(\mathbf{x}) - E_{k-1}(\mathbf{x})|}{E_k(\mathbf{x})} \tag{31}$$

where $E(\mathbf{x})$ is the value of the objective function at $\mathbf{x}$ and k is the iteration number. A suitable termination tolerance was found to be $\epsilon \leq 1 \times 10^{-7}$.

VI. Comparison of the Polynomial and Polar Formulations

On the basis of results obtained in a large number of designs satisfying a variety of specifications, the following general observations can be made:

a) Although the polar formulation requires the evaluation of transcendental functions, it usually requires *less* computational effort than the polynomial formulation. This is because it has simpler constraints and also typically fewer iterations are required to achieve convergence.

b) The polar formulation usually outperforms the polynomial formulation especially with respect to group-delay performance. However, occasionally a better magnitude response is achieved with the polynomial formulation.

VII. Design Examples

In this section, we provide detailed results obtained in four design examples, each intended to demonstrate some feature of the proposed methodology. In all designs a value of $p = 2$ was assumed to facilitate comparisons with competing designs, i.e., all designs were essentially least-squares solutions.

The frequencies ω_i in (6) were selected using the nonuniform adaptive sampling technique described in [19], which we have found to be very effective in eliminating spikes in the error function. This technique was implemented as an integral part of the optimization algorithm. The underlying principles involved are as follows: Each passband and stopband is segmented into a fixed number of K frequency intervals, possibly nonuniform, and fixed sample frequencies are assigned in the first and last intervals, usually at the band edges. Before each iteration, the error in each band is evaluated over a dense grid of L so-called *virtual sample frequencies* where L is typically at least $10K$ and the frequency of maximum error in each of the remaining $K-2$ intervals is located and then used as a sample frequency. In this way, *frequency points at which spikes are beginning to form in a given iteration are located and are used as sample frequencies in the next iteration*. Transition bands are treated as a '*don't-care*' bands, i.e., no sample frequencies are assigned. (See [19] for details).

In the present experiments, which involved the design of lowpass filters, fixed sample frequencies were assigned in the first and last three passband intervals and the first three and last stopband intervals. The weights used for these fixed sample frequencies were 1.0, 0.8. 0.6, 0.5, 0.4, 0.4, 0.5, 0.6, 0.8, 1.0, from left to right. The weights used for the variable sample frequencies were all set to unity.

The numerical values of the parameters used in the nonuniform sampling technique were as follows:

- sampling frequency $\omega_s = 2\pi$ rad/s;
- initial sample frequency $\omega_{\text{init}} = 0.001$ rad/s;
- final sample frequency $= \omega_s/2 - \omega_{\text{init}} = 3.1406$ rad/s;
- number of virtual sample points for each band $L = 1004$;
- number of sample frequencies (variable plus fixed) $K = 34$ in Example 1 and 24 in each of Examples 2–4.

TABLE I
Lowpass Digital-Filter Specifications for Example Using an Initial Design Obtained With the Method of Trends (Example 1)

Parameter	Value
Maximum passband ripple A_p, dB	0.1
Minimum stopband attenuation, A_a,dB	40
Passband edge ω_p, rad/s	0.2π
Stopband edge ω_a, rad/s	0.3π
Stand. dev. in group delay, %	6

TABLE II
Design Parameters for Example 1

Parameter	Polyn.	Polar
Stability margin δ_s	0.05	0.04
Real pole boundary margin δ_m	0.05	0.05
Initial step limit β_{c_i}	0.03	0.05
Initial step limit β_{H_0}	0.03	0.05
Initial step limit β_τ	0.1	0.1
Initial delay τ_0, s	17	18

In Example 1, we designed a filter that would satisfy the specifications in Table I using the polynomial and polar formulations along with the method of trends described in [22] for the initial design. The design parameters and results obtained are listed in Tables II and III, respectively. The fixed sample frequencies were 0.001, 0.6264, 0.6271, 0.6277, 0.6283, 0.9425, 0.9447, 0.9469, 0.9491, 3.1406. As can be seen in Table III, the polynomial formulation resulted in slightly better stopband performance whereas the polar formulation resulted in a reduced group-delay variation and slightly better passband magnitude response.

In view of the fact that the polar formulation usually offers certain advantages over the polynomial formation, as detailed in Section VI, we have used the polar formulation for our designs of the filters in Examples 2 to 4 along with the BMT method for the initialization.

In Example 2, we designed a filter that would satisfy the specifications in Table IV and compared it with a corresponding filter obtained by using an elliptic filter in cascade with a delay equalizer [2]. The initial design was generated using a nonrecursive filter of order 33. The fixed sample frequencies were 0.001, 1.8793, 1.8812, 1.8831, 1.8850, 2.1991, 2.2001, 2.2010 2.2019, 3.1406. The design parameters and results obtained are listed in Tables V and VI. As can be seen, the proposed method produced a filter with a significantly lower order and improved performance, especially reduced group delay.

In Example 3, we designed a filter that would satisfy the specifications in Table VII. In this case, the initial design was obtained by using a 25th-order nonrecursive filter. The fixed sample frequencies were 0.001, 1.5661, 1.5677, 1.5692, 1.5708, 1.8850, 1.8862, 1.8875, 1.8887, 3.1406. The design parameters and results obtained are listed in Tables VIII and IX. In Table IX we also show results obtained by using the cone-quadratic-programming (CQP) method proposed in [27]. Once again, we note that the proposed method resulted in an improved design with a notably lower passband ripple and group delay. For consistency

TABLE III
DESIGN RESULTS FOR EXAMPLE 1

	Polynomial	Polar
Filter order n	10	10
Optimization iterations	181	81
Elapsed CPU time, s	99.52	39.42
Maximum passband ripple, dB	0.0689	0.0637
Minimum stopband attenuation, dB	43.15	40.23
Passband group delay, s	16.78	17.78
Stand. dev. of passband group delay, %	5.96	3.27
Final value of objective function $E(\mathbf{x})$	7.4783 $\times 10^{-4}$	1.0447 $\times 10^{-3}$

TABLE IV
LOWPASS DIGITAL-FILTER SPECIFICATIONS FOR EXAMPLE COMPARING THE PROPOSED AND EQUALIZER METHODS (EXAMPLE 2)

Parameter	Value
Maximum passband ripple A_p, dB	0.05
Minimum stopband attenuation, A_a,dB	45
Passband edge ω_p, rad/s	0.6π
Stopband edge ω_a, rad/s	0.7π
Stand. dev. in group delay, %	6

TABLE V
DESIGN PARAMETERS FOR EXAMPLE 2

Parameter	Value
Stability margin δ_s	0.08
Initial step limit $\beta_{\hat{p}_i}$	0.03
Initial step limit β_{H_0}	0.03
Initial step limit β_τ	0.1

TABLE VI
DESIGN RESULTS FOR EXAMPLE 2

	Equalizer	Proposed
Elliptic filter order	6	NA
Equalizer filter order	22	NA
Total filter order	28	20
Elapsed CPU time, s	0.44	92.36
Maximum passband ripple, dB	0.0500	0.0444
Minimum stopband attenuation, dB	45.00	46.24
Passband group delay, s	34.84	17.08
Stand. dev. of passband group delay, %	4.26	5.47
L_2 Norm of the error	4.6477	7.4099

TABLE VII
LOWPASS DIGITAL-FILTER SPECIFICATIONS FOR EXAMPLE COMPARING THE PROPOSED AND CQP METHODS (EXAMPLE 3)

Parameter	Value
Maximum passband ripple A_p, dB	0.35
Minimum stopband attenuation, A_a, dB	34
Passband edge ω_p, rad/s	0.5π
Stopband edge ω_a, rad/s	0.6π
Stand., dev. in group delay, %	6

TABLE VIII
DESIGN PARAMETERS FOR EXAMPLE 3

Parameter	Value
Stability margin, δ_s	0.1
Initial step limit $\beta_{\hat{p}_i}$	0.02
Initial step limit β_{H_0}	0.02
Initial step limit β_τ	0.04

TABLE IX
DESIGN RESULTS FOR EXAMPLE 3

Objective Function	CQP	Proposed
Filter order n	12	12
Maximum passband ripple, dB	0.266	0.126
Minimum stopband attenuation, dB	36.15	38.86
Passband group delay, s	15.90	11.91
L_2 norm of the error	5.2865	7.6329
Average value of τ_{ri}	0.00874	0.00364
Maximum value of τ_{ri}	0.06947	0.06794

TABLE X
LOWPASS DIGITAL-FILTER SPECIFICATIONS FOR THE EXAMPLE COMPARING PARAMETERIZED AND PROPOSED METHODS (EXAMPLE 4)

Parameter	Value
Maximum passband ripple A_p, dB	0.1
Minimum stopband attenuation, A_a, dB	45
Passband edge, ω_a, rad/s	0.4π
Stopband edge, ω_p, rad/s	0.6π
Maximum relative deviation of group delay in passband, %	5

TABLE XI
DESIGN PARAMETERS FOR EXAMPLE 4

Parameter	Value
Stability margin, δ_s	0.05
Initial step limit $\beta_{\hat{p}_i}$	0.01
Initial step limit β_{H_0}	0.01
Initial step limit β_τ	0.02

with [27], Table IX includes the average and maximum values of the relative deviation of the passband group delay τ_{ri}, which is defined as

$$\tau_{ri} = \frac{|\tau_i - \overline{\tau}|}{\overline{\tau}} \tag{32}$$

where τ_i is the group delay at frequency $\omega_i \in \mathbf{\Omega}$ and $\overline{\tau}$ is the average group delay across the passband.

In Example 4, we compare our method with an *unconstrained* optimization method proposed in [21]. In this approach, the filter transfer function is parameterized using a hyperbolic tangent transformation (HTT), an arctangent transformation (ATT), or a modified bilinear transformation (MBT). The resulting optimization problem is then solved by using either the quasi-Newton method of Broyden, Fletcher, Goldfarb, and Shanno (BFGS) or a modified Newton's method. The filter specifications and corresponding design parameters for our method are given in Tables X and XI. The initial transfer function was obtained by using a 16th-order nonrecursive filter. The fixed sample frequencies were 0.001, 1.2529, 1.2541, 1.2554 1.2566, 1.8850, 1.8862, 1.8875, 1.8887, 3.1406. In Table XII, we compare our results with those reported in [21]. As can be seen, the proposed method gave a reduced passband

TABLE XII
DESIGN RESULTS FOR EXAMPLE 4

Optimization algorithm	BFGS			Modified Newton			Proposed
Parameterization	HTT	ATT	MBT	HTT	ATT	MBT	
Filter order n	10	10	10	10	10	10	10
Number of iterations	99	164	126	74	91	90	37
Maximum passband ripple, dB	0.0818	0.0663	0.0311	0.0773	0.0389	0.0785	0.0233
Minimum stopband attenuation, dB	47.52	49.31	46.52	48.46	45.74	46.98	50.95
Max. rel. deviation of τ in passband, %	4.19	3.63	4.90	1.89	4.78	1.80	1.79

ripple as well as reduced group-delay variation and, in addition, increased minimum stopband attenuation.

We are currently in the process of implementing our algorithm in terms of the least-pth algorithm described in [20] which is expected to yield reduced magnitude-response error as well as reduced group-delay error. In addition, we plan to compare our methodology with some of the more promising of the recent methods referenced in the introduction. An interesting approach to explore would be to add one or more nonrecursive sections to the cascade arrangement. In this way, additional arbitrary zeros, which are accompanied by corresponding poles at the origin of the z plane, can be introduced. As stated by Lang in [9], zeros tend to exert significant influence on the frequency response with respect to passbands where phase response linearity is required and, in effect, a set of nonrecursive sections can imitate the behavior of a phase equalizer. In fact, by this means, Lang was able to obtain better lowpass filters than corresponding filters reported in the past by Descky in [28] and Lu in [29]. We have also obtained a better lowpass filter than one reported by Lu and Hinamoto in [27] but the last word has not been said on this issue. Further comparisons need to be carried out with the conic-programming approach as well as with some of the other competing methods over a diverse range of filter types including bandpass and bandstop filters not to mention differentiators and Hilbert transformers.

One of the reviewers expressed the opinion that the case of cascaded higher order sections should also have been considered. Unfortunately, when the order of the filter sections is increased above two, the simplicity of the stability constraints as well as the real-pole boundary constraints would be destroyed and essentially our methodology would be difficult if not impossible to apply. A consequence of that would be that the optimization problem at hand would no longer fit the mold of the standard quadratic programming problem in (5). Nevertheless, the use of higher-order transfer functions could be explored by using Lang's method in [9] whereby the stability issue is handled in an indirect way using Rouché's theorem. Further work comparing a diverse range of designs carried out with the methodology as presented in this paper and designs carried out with Lang's approach would resolve the issue as to whether one or more higher-order sections would be better than a set of cascaded second-order sections.

VIII. CONCLUSION

A methodology for the design of recursive digital filters with nearly linear phase response has been described. The design problem is formulated as a constrained quadratic programming problem and is solved using an optimization engine based on Newton's method. Two different formulations of the problem have been explored, namely, the polynomial and polar formulations. A number of practical aspects associated with the design of recursive filters in general have also been investigated such as initialization and the handling of stability and step-size constraints to achieve stability on the one hand and to expedite convergence on the other. Furthermore, a technique that can be used to circumvent a real-axis pole-migration problem inherent in the design of recursive filters by optimization has been proposed. Many design examples have revealed that the polar formulation usually outperforms the polynomial formulation and, in addition, it requires reduced computational effort. Relative to designs based on some of the existing state-of-the-art approaches, e.g., the classical equalizer approach, the proposed methodology yields filters having significantly lower order and/or improved performance.

APPENDIX A
GRADIENT AND HESSIAN FOR THE POLYNOMIAL FORMULATION

A. Gradient of the Error Function

The gradient of the error function is given by

$$\nabla e_i(\mathbf{x}) = \left[\frac{\partial e_i(\mathbf{x})}{\partial a_{01}}\ \frac{\partial e_i(\mathbf{x})}{\partial a_{11}}\ \frac{\partial e_i(\mathbf{x})}{\partial b_{01}}\ \frac{\partial e_i(\mathbf{x})}{\partial b_{11}} \cdots \right. \left. \frac{\partial e_i(\mathbf{x})}{\partial b_{0J}}\ \frac{\partial e_i(\mathbf{x})}{\partial b_{1J}}\ \frac{\partial e_i(\mathbf{x})}{\partial H_0}\ \frac{\partial e_i(\mathbf{x})}{\partial \tau}\right]^T \tag{33}$$

where

$$\begin{aligned}
\frac{\partial e_i(\mathbf{x})}{\partial a_{0k}} &= \frac{F(\mathbf{x},\omega_i)}{N_k(e^{j\omega_i T})}\\
\frac{\partial e_i(\mathbf{x})}{\partial a_{1k}} &= \frac{e^{j\omega_i T}F(\mathbf{x},\omega_i)}{N_k(e^{j\omega_i T})}\\
\frac{\partial e_i(\mathbf{x})}{\partial b_{0k}} &= -\frac{F(\mathbf{x},\omega_i)}{D_k(e^{j\omega_i T})}\\
\frac{\partial e_i(\mathbf{x})}{\partial b_{1k}} &= -\frac{e^{j\omega_i T}F(\mathbf{x},\omega_i)}{D_k(e^{j\omega_i T})}\\
\frac{\partial e_i(\mathbf{x})}{\partial H_0} &= \frac{F(\mathbf{x},\omega_i)}{H_0}\\
\frac{\partial e_i(\mathbf{x})}{\partial \tau} &= j\omega_i F_0(\omega_i).
\end{aligned}$$

Although the derivatives of the error function are complex quantities, (8) yields a real-valued gradient since the imaginary parts cancel out when the errors and their conjugates are added.

B. Hessian of the Error Function

The Hessian of the error function, $\nabla^2 e_i(\mathbf{x})$, is defined as

$$\nabla^2 e_i(\mathbf{x}) = \begin{bmatrix} \frac{\partial e_i^2(\mathbf{x})}{\partial a_{01}\partial a_{01}} & \frac{\partial e_i^2(\mathbf{x})}{\partial a_{01}\partial a_{11}} & \cdots & \frac{\partial e_i^2(\mathbf{x})}{\partial a_{01}\partial H_0} & \frac{\partial e_i^2(\mathbf{x})}{\partial a_{01}\partial \tau} \\ \frac{\partial e_i^2(\mathbf{x})}{\partial a_{11}\partial a_{01}} & \frac{\partial e_i^2(\mathbf{x})}{\partial a_{11}\partial a_{11}} & \cdots & \frac{\partial e_i^2(\mathbf{x})}{\partial a_{11}\partial H_0} & \frac{\partial e_i^2(\mathbf{x})}{\partial a_{11}\partial \tau} \\ \frac{\partial e_i^2(\mathbf{x})}{\partial b_{01}\partial a_{01}} & \frac{\partial e_i^2(\mathbf{x})}{\partial b_{01}\partial a_{11}} & \cdots & \frac{\partial e_i^2(\mathbf{x})}{\partial b_{01}\partial H_0} & \frac{\partial e_i^2(\mathbf{x})}{\partial b_{01}\partial \tau} \\ \frac{\partial e_i^2(\mathbf{x})}{\partial b_{11}\partial a_{01}} & \frac{\partial e_i^2(\mathbf{x})}{\partial b_{11}\partial a_{11}} & \cdots & \frac{\partial e_i^2(\mathbf{x})}{\partial b_{11}\partial H_0} & \frac{\partial e_i^2(\mathbf{x})}{\partial b_{11}\partial \tau} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{\partial e_i^2(\mathbf{x})}{\partial H_0\partial a_{01}} & \frac{\partial e_i^2(\mathbf{x})}{\partial H_0\partial a_{11}} & \cdots & \frac{\partial e_i^2(\mathbf{x})}{\partial H_0\partial H_0} & \frac{\partial e_i^2(\mathbf{x})}{\partial H_0\partial \tau} \\ \frac{\partial e_i^2(\mathbf{x})}{\partial \tau\partial a_{01}} & \frac{\partial e_i^2(\mathbf{x})}{\partial \tau\partial a_{11}} & \cdots & \frac{\partial e_i^2(\mathbf{x})}{\partial \tau\partial H_0} & \frac{\partial e_i^2(\mathbf{x})}{\partial \tau\partial \tau} \end{bmatrix} \tag{34}$$

Unfortunately, this matrix is not block diagonal as in the equalizer case [16] where the mirror-image numerator and denominator polynomials give rise to certain symmetry properties in the derivatives of the error function.

The Hessian in (34) can be simplified by splitting it into several block-symmetric matrices and then taking advantage of its inherent symmetry properties. The second-order partial derivatives with respect to the transfer function coefficients are divided into two block matrix types, diagonal 4×4 block matrices and lower triangular 4×4 block matrices. The diagonal block matrices are given by

$$H_{mm}^{(\mathrm{diag})} = \begin{bmatrix} 0 & 0 & \frac{\partial e_i^2(\mathbf{x})}{\partial a_{0m}\partial b_{0m}} & \frac{\partial e_i^2(\mathbf{x})}{\partial a_{0m}\partial b_{1m}} \\ 0 & 0 & \frac{\partial e_i^2(\mathbf{x})}{\partial a_{1m}\partial b_{0m}} & \frac{\partial e_i^2(\mathbf{x})}{\partial a_{1m}\partial b_{1m}} \\ \frac{\partial e_i^2(\mathbf{x})}{\partial b_{0m}\partial a_{0m}} & \frac{\partial e_i^2(\mathbf{x})}{\partial b_{0m}\partial a_{1m}} & \frac{\partial e_i^2(\mathbf{x})}{\partial b_{0m}\partial b_{0m}} & \frac{\partial e_i^2(\mathbf{x})}{\partial b_{0m}\partial b_{1m}} \\ \frac{\partial e_i^2(\mathbf{x})}{\partial b_{1m}\partial a_{0m}} & \frac{\partial e_i^2(\mathbf{x})}{\partial b_{1m}\partial a_{1m}} & \frac{\partial e_i^2(\mathbf{x})}{\partial b_{1m}\partial b_{0m}} & \frac{\partial e_i^2(\mathbf{x})}{\partial b_{1m}\partial b_{1m}} \end{bmatrix} \tag{35}$$

where $m = 1, \ldots, J$ is the section number. The $H_{mm}^{(\mathrm{diag})}$ block matrices are the symmetric blocks that are located along the diagonal of the Hessian and represent all the second-order partial derivative combinations with respect to the transfer function coefficients for the same section. By virtue of the symmetry property of the Hessian, the upper triangular matrix is equal to the lower triangular matrix. Therefore, only the equations for the lower triangular matrix and the diagonal second-order partial derivatives need to be evaluated. It was also found that the upper 2×2 block diagonal submatrices of the $H_{mm}^{(\mathrm{diag})}$ blocks are equal to zero. The equations for the $H_{mm}^{(\mathrm{diag})}$ block matrices are

$$\frac{\partial e_i^2(\mathbf{x})}{\partial b_{0m}\partial a_{0m}} = -\frac{F(\mathbf{x},\omega_i)}{D_j(e^{j\omega_i T})N_m(e^{j\omega_i T})} \tag{36}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial b_{1m}\partial a_{0m}} = -\frac{e^{j\omega_i T}F(\mathbf{x},\omega_i)}{D_m(e^{j\omega_i T})N_m(e^{j\omega_i T})} \tag{37}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial b_{1m}\partial a_{1m}} = -\frac{e^{j2\omega_i T}F(\mathbf{x},\omega_i)}{D_m(e^{j\omega_i T})N_m(e^{j\omega_i T})} \tag{38}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial b_{0m}\partial b_{0m}} = \frac{2F(\mathbf{x},\omega_i)}{D_m^2(e^{j\omega_i T})} \tag{39}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial b_{1m}\partial b_{0m}} = \frac{2e^{j\omega_i T}F(\mathbf{x},\omega_i)}{D_m^2(e^{j\omega_i T})} \tag{40}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial b_{1m}\partial b_{1m}} = \frac{2e^{j2\omega_i T}F(\mathbf{x},\omega_i)}{D_m^2(e^{j\omega_i T})} \tag{41}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial b_{0m}\partial a_{1m}} = \frac{\partial e_i^2(\mathbf{x})}{\partial b_{1m}\partial a_{0m}} \tag{42}$$

As can be seen in (42), all the 2×2 submatrices within the $H_{mm}^{(\mathrm{diag})}$ blocks are block symmetric and, therefore, only 6 equations are needed to obtain all the $H_{mm}^{(\mathrm{diag})}$ blocks in the Hessian.

The lower triangular blocks are given by

$$H_{jm}^{(\mathrm{lower})} = \begin{bmatrix} \frac{\partial e_i^2(\mathbf{x})}{\partial a_{0j}\partial a_{0m}} & \frac{\partial e_i^2(\mathbf{x})}{\partial a_{0j}\partial a_{1m}} & \frac{\partial e_i^2(\mathbf{x})}{\partial a_{0j}\partial b_{0m}} & \frac{\partial e_i^2(\mathbf{x})}{\partial a_{0j}\partial b_{1m}} \\ \frac{\partial e_i^2(\mathbf{x})}{\partial a_{1j}\partial a_{0m}} & \frac{\partial e_i^2(\mathbf{x})}{\partial a_{1j}\partial a_{1m}} & \frac{\partial e_i^2(\mathbf{x})}{\partial a_{1j}\partial b_{0m}} & \frac{\partial e_i^2(\mathbf{x})}{\partial a_{1j}\partial b_{1m}} \\ \frac{\partial e_i^2(\mathbf{x})}{\partial b_{0j}\partial a_{0m}} & \frac{\partial e_i^2(\mathbf{x})}{\partial b_{0j}\partial a_{1m}} & \frac{\partial e_i^2(\mathbf{x})}{\partial b_{0j}\partial b_{0m}} & \frac{\partial e_i^2(\mathbf{x})}{\partial b_{0j}\partial b_{1m}} \\ \frac{\partial e_i^2(\mathbf{x})}{\partial b_{1j}\partial a_{0m}} & \frac{\partial e_i^2(\mathbf{x})}{\partial b_{1j}\partial a_{1m}} & \frac{\partial e_i^2(\mathbf{x})}{\partial b_{1j}\partial b_{0m}} & \frac{\partial e_i^2(\mathbf{x})}{\partial b_{1j}\partial b_{1m}} \end{bmatrix} \tag{43}$$

where $j = 1, \ldots, J$, $m = 1, \ldots, J$, $j \neq m$ are the transfer function section numbers. Unlike the diagonal block matrices, the lower triangular block matrices are not symmetric and, therefore, there are 16 equations for each block. The equations for each of the four columns of the $H_{jm}^{(\mathrm{lower})}$ block matrices follow.

First column:

$$\frac{\partial e_i^2(\mathbf{x})}{\partial a_{0j}\partial a_{0m}} = \frac{F(\mathbf{x},\omega_i)}{N_j(e^{j\omega_i T})N_m(e^{j\omega_i T})}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial a_{1m}\partial a_{0m}} = \frac{e^{j\omega_i T}F(\mathbf{x},\omega_i)}{N_j(e^{j\omega_i T})N_m(e^{j\omega_i T})}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial b_{0j}\partial a_{0m}} = -\frac{F(\mathbf{x},\omega_i)}{D_j(e^{j\omega_i T})N_m(e^{j\omega_i T})}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial b_{1j}\partial a_{0m}} = -\frac{e^{j\omega_i T}F(\mathbf{x},\omega_i)}{D_j(e^{j\omega_i T})N_m(e^{j\omega_i T})}$$

Second column:

$$\frac{\partial e_i^2(\mathbf{x})}{\partial a_{0j}\partial a_{1m}} = \frac{e^{j\omega_i T}F(\mathbf{x},\omega_i)}{N_j(e^{j\omega_i T})N_m(e^{j\omega_i T})}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial a_{1j}\partial a_{1m}} = \frac{e^{j2\omega_i T}F(\mathbf{x},\omega_i)}{N_j(e^{j\omega_i T})N_m(e^{j\omega_i T})}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial b_{0j}\partial a_{1m}} = -\frac{e^{j\omega_i T}F(\mathbf{x},\omega_i)}{D_j(e^{j\omega_i T})N_m(e^{j\omega_i T})}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial b_{1j}\partial a_{1m}} = -\frac{e^{j2\omega_i T}F(\mathbf{x},\omega_i)}{D_j(e^{j\omega_i T})N_m(e^{j\omega_i T})}$$

Third column:

$$\frac{\partial e_i^2(\mathbf{x})}{\partial a_{0j}\partial b_{0m}} = -\frac{F(\mathbf{x},\omega_i)}{N_j(e^{j\omega_i T})D_m(e^{j\omega_i T})}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial a_{1j}\partial b_{0m}} = -\frac{e^{j\omega_i T}F(\mathbf{x},\omega_i)}{N_j(e^{j\omega_i T})D_m(e^{j\omega_i T})}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial b_{0j}\partial b_{0m}} = \frac{F(\mathbf{x},\omega_i)}{D_j(e^{j\omega_i T})D_m(e^{j\omega_i T})}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial b_{1j}\partial b_{0m}} = \frac{e^{j\omega_i T}F(\mathbf{x},\omega_i)}{D_j(e^{j\omega_i T})D_m(e^{j\omega_i T})}$$

Fourth column:

$$\frac{\partial e_i^2(\mathbf{x})}{\partial a_{0j}\partial b_{1m}} = -\frac{e^{j\omega_i T}F(\mathbf{x},\omega_i)}{N_j(e^{j\omega_i T})D_m(e^{j\omega_i T})}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial a_{1j}\partial b_{1m}} = -\frac{e^{j2\omega_i T}F(\mathbf{x},\omega_i)}{N_j(e^{j\omega_i T})D_m(e^{j\omega_i T})}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial b_{0j}\partial b_{1m}} = \frac{e^{j\omega_i T}F(\mathbf{x},\omega_i)}{D_j(e^{j\omega_i T})D_m(e^{j\omega_i T})}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial b_{1j}\partial b_{1m}} = \frac{e^{j2\omega_i T}F(\mathbf{x},\omega_i)}{D_j(e^{j\omega_i T})D_m(e^{j\omega_i T})}$$

The remaining equations needed to construct the Hessian are the second-order partial derivatives with respect to multiplier constant, H_0, and group delay, τ. The second-order partial derivatives with respect to H_0 are

$$\begin{aligned}
\frac{\partial e_i^2(\mathbf{x})}{\partial H_0 \partial a_{0j}} &= \frac{F(\mathbf{x}, \omega_i)}{H_0 N_m(e^{j\omega_i T})} \\
\frac{\partial e_i^2(\mathbf{x})}{\partial H_0 \partial a_{1j}} &= \frac{e^{j\omega_i T} F(\mathbf{x}, \omega_i)}{H_0 N_m(e^{j\omega_i T})} \\
\frac{\partial e_i^2(\mathbf{x})}{\partial H_0 \partial b_{0j}} &= -\frac{F(\mathbf{x}, \omega_i)}{H_0 D_m(e^{j\omega_i T})} \\
\frac{\partial e_i^2(\mathbf{x})}{\partial H_0 \partial b_{1j}} &= -\frac{e^{j\omega_i T} F(\mathbf{x}, \omega_i)}{H_0 D_m(e^{j\omega_i T})}
\end{aligned}$$

those of the filter coefficients with respect to the group delay, τ, are all zero, i.e.,

$$\frac{\partial e_i^2(\mathbf{x})}{\partial \tau \partial a_{0j}} = \frac{\partial e_i^2(\mathbf{x})}{\partial \tau \partial a_{1j}} = \frac{\partial e_i^2(\mathbf{x})}{\partial \tau \partial b_{0j}} = \frac{\partial e_i^2(\mathbf{x})}{\partial \tau \partial b_{1j}} = 0$$

and the remaining derivatives are given by

$$\begin{aligned}
&\frac{\partial e_i^2(\mathbf{x})}{\partial H_0 \partial H_0} = 0, \quad \frac{\partial e_i^2(\mathbf{x})}{\partial \tau \partial H_0} = 0, \\
\text{and} \quad &\frac{\partial e_i^2(\mathbf{x})}{\partial \tau \partial \tau} = -\,\omega_i^2 M_0(\omega_i) e^{-j\omega_i \tau}.
\end{aligned}$$

Appendix B
Constraints for the Polynomial Formulation

The constraints are formulated as

$$\mathbf{A}\boldsymbol{\delta} \leq \mathbf{b}$$

where each element of $\boldsymbol{\delta}$ is the allowed change in the corresponding element of the parameter vector $\mathbf{x}$

A. Step-Size Constraints

Letting $\delta_{a ji}$ and $\delta_{b ji}$ be equal to the allowed changes in the respective filter coefficients and δ_{H_0} and δ_τ be equal to the allowed changes in H_0 and τ, respectively, during each iteration, the inequality

$$\mathbf{A}_1 \boldsymbol{\delta}_1 \leq \mathbf{b}_1 \tag{44}$$

assumes the form

$$\begin{bmatrix}
1 & 0 & \cdots & 0 & 0 & 0 \\
-1 & 0 & \cdots & 0 & 0 & 0 \\
0 & 1 & \cdots & 0 & 0 & 0 \\
0 & -1 & \cdots & 0 & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
0 & 0 & \cdots & 1 & 0 & 0 \\
0 & 0 & \cdots & -1 & 0 & 0 \\
0 & 0 & \cdots & 0 & 1 & 0 \\
0 & 0 & \cdots & 0 & -1 & 0 \\
0 & 0 & \cdots & 0 & 0 & 1 \\
0 & 0 & \cdots & 0 & 0 & -1
\end{bmatrix}
\begin{bmatrix}
\delta_{a_{01}}^{(k)} \\ \delta_{a_{11}}^{(k)} \\ \delta_{b_{01}}^{(k)} \\ \delta_{b_{11}}^{(k)} \\ \vdots \\ \delta_{b_{0J}}^{(k)} \\ \delta_{b_{1J}}^{(k)} \\ \delta_{H_0}^{(k)} \\ \delta_{H_0}^{(k)} \\ \delta_{\tau}^{(k)} \\ \delta_{\tau}^{(k)}
\end{bmatrix}
\leq
\begin{bmatrix}
\beta_c^{(k)} \\ \beta_c^{(k)} \\ \beta_c^{(k)} \\ \beta_c^{(k)} \\ \vdots \\ \beta_c^{(k)} \\ \beta_c^{(k)} \\ \beta_{H_0}^{(k)} \\ \beta_{H_0}^{(k)} \\ \beta_{\tau}^{(k)} \\ \beta_{\tau}^{(k)}
\end{bmatrix}.$$

B. Stability Constraints

The stability constraints can be expressed in terms of $\boldsymbol{\delta}$ as

$$\mathbf{A}_2 \boldsymbol{\delta}_2 \leq \mathbf{b}_2 \tag{45}$$

and since

$$b_{0i}^{(k)} = b_{0i}^{(k-1)} + \delta_{b_{0i}}^{(k)} \tag{46}$$

$$b_{1i}^{(k)} = b_{1i}^{(k-1)} + \delta_{b_{1i}}^{(k)} \tag{47}$$

we get

$$\delta_{b_{0i}}^{(k)} \leq 1 - \delta_s - b_{0i}^{(k-1)} \tag{48}$$

$$\delta_{b_{1i}}^{(k)} - \delta_{b_{0i}}^{(k)} \leq 1 - \delta_s - b_{1i}^{(k-1)} + b_{0i}^{(k-1)} \tag{49}$$

$$-\delta_{b_{1i}}^{(k)} - \delta_{b_{0i}}^{(k)} \leq 1 - \delta_s + b_{1i}^{(k-1)} + b_{0i}^{(k-1)}. \tag{50}$$

Alternatively, the stability constraints can be expressed in matrix form as

$$\begin{bmatrix}
\mathbf{A}_2 & \mathbf{0} & \cdots & \mathbf{0} \\
\mathbf{0} & \mathbf{A}_2 & \cdots & \mathbf{0} \\
\vdots & \vdots & \ddots & \vdots \\
\mathbf{0} & \mathbf{0} & \cdots & \mathbf{A}_2
\end{bmatrix}
\begin{bmatrix}
\delta_{b_{01}}^{(k)} \\ \delta_{b_{11}}^{(k)} \\ \vdots \\ \delta_{b_{1J}}^{(k)}
\end{bmatrix}
\leq
\begin{bmatrix}
\mathbf{b}_2^{(1,k)} \\ \mathbf{b}_2^{(2,k)} \\ \vdots \\ \mathbf{b}_2^{(J,k)}
\end{bmatrix} \tag{51}$$

where

$$\mathbf{A}_2 = \begin{bmatrix} 1 & 0 \\ -1 & 1 \\ -1 & -1 \end{bmatrix}$$

$$\mathbf{b}_2^{(i,k)} = \begin{bmatrix} 1 - \delta_s - b_{0i}^{(k-1)} \\ 1 - \delta_s - b_{1i}^{(k-1)} + b_{0i}^{(k-1)} \\ 1 - \delta_s + b_{1i}^{(k-1)} + b_{0i}^{(k-1)} \end{bmatrix}.$$

C. Real-Pole Boundary Constraints

The real-pole boundary constraints can be expressed in terms of $\boldsymbol{\delta}$ as

$$\mathbf{A}_3 \boldsymbol{\delta}_3 \leq \mathbf{b}_3 \tag{52}$$

Equations (22)–(28) have their tangent points on the real-pole boundary. Because the tangent-line boundary in Fig. 2 extends a small amount into the real-pole region, we introduce a margin-control variable δ_m that shifts the real-pole boundary in Fig. 2 to the right. Expressing the constraint in terms of $\boldsymbol{\delta}$ and including δ_m in (22)–(28), we get

$$\delta_{b_{1i}}^{(k)} - 2\delta_{b_{0i}}^{(k)} \leq \frac{1}{2} - \delta_m + 2b_{0i}^{(k-1)} - b_{1i}^{(k-1)} \tag{53}$$

$$\delta_{b_{1i}}^{(k)} - \frac{4}{3}\delta_{b_{0i}}^{(k)} \leq \frac{3}{4} - \delta_m + \frac{4}{3}b_{0i}^{(k-1)} - b_{1i}^{(k-1)} \tag{54}$$

$$\delta_{b_{1i}}^{(k)} - 4\delta_{b_{0i}}^{(k)} \leq \frac{1}{4} - \delta_m + 4b_{0i}^{(k-1)} - b_{1i}^{(k-1)} \tag{55}$$

$$-\delta_{b_{0i}}^{(k)} \leq -\delta_m + b_{0i}^{(k-1)} \tag{56}$$

$$-\delta_{b_{1i}}^{(k)} - 2\delta_{b_{0i}}^{(k)} \leq \frac{1}{2} - \delta_m + 2b_{0i}^{(k-1)} + b_{1i}^{(k-1)} \tag{57}$$

$$-\delta_{b_{1i}}^{(k)} - \frac{4}{3}\delta_{b_{0i}}^{(k)} \leq \frac{3}{4} - \delta_m + \frac{4}{3}b_{0i}^{(k-1)} + b_{1i}^{(k-1)} \tag{58}$$

$$-\delta_{b_{1i}}^{(k)} - 4\delta_{b_{0i}}^{(k)} \leq \frac{1}{4} - \delta_m + 4b_{0i}^{(k-1)} + b_{1i}^{(k-1)}. \tag{59}$$

The constraint equations in (52) can be expressed as

$$\begin{bmatrix} \mathbf{A}_3 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_3 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{A}_3 \end{bmatrix} \begin{bmatrix} \delta_{b_{01}}^{(k)} \\ \delta_{b_{11}}^{(k)} \\ \vdots \\ \delta_{b_{1i}}^{(k)} \end{bmatrix} \le \begin{bmatrix} \mathbf{b}_3^{(1,k)} \\ \mathbf{b}_3^{(2,k)} \\ \vdots \\ \mathbf{b}_3^{(J,k)} \end{bmatrix} \tag{60}$$

where

$$\mathbf{A}_3 = \begin{bmatrix} -2 & 1 \\ -\frac{4}{3} & 1 \\ -4 & 1 \\ -1 & 0 \\ -2 & -1 \\ -\frac{4}{3} & -1 \\ -4 & -1 \end{bmatrix}$$

$$\mathbf{b}_3^{(i,k)} = \begin{bmatrix} \frac{1}{2} - \delta_m + 2b_{0i}^{(k-1)} - b_{1i}^{(k-1)} \\ \frac{3}{4} - \delta_m + \frac{4}{3}b_{0i}^{(k-1)} - b_{1i}^{(k-1)} \\ \frac{1}{4} - \delta_m + 4b_{0i}^{(k-1)} - b_{1i}^{(k-1)} \\ -\delta_m + b_{0i}^{(k-1)} \\ \frac{1}{2} - \delta_m + 2b_{0i}^{(k-1)} + b_{1i}^{(k-1)} \\ \frac{3}{4} - \delta_m + \frac{4}{3}b_{0i}^{(k-1)} + b_{1i}^{(k-1)} \\ \frac{1}{4} - \delta_m + 4b_{0i}^{(k-1)} + b_{1i}^{(k-1)} \end{bmatrix}.$$

Appendix C
Gradient and Hessian For the Polar Formulation

A. Gradient of the Error Function

The gradient of the error function is given by

$$\nabla e_i(\mathbf{x}) = \left[\frac{\partial e_i(\mathbf{x})}{\partial r_{a1}} \; \frac{\partial e_i(\mathbf{x})}{\partial \theta_{a1}} \; \frac{\partial e_i(\mathbf{x})}{\partial r_{b1}} \; \frac{\partial e_i(\mathbf{x})}{\partial \theta_{b1}} \cdots \right.$$
$$\left. \frac{\partial e_i(\mathbf{x})}{\partial r_{bJ}} \; \frac{\partial e_i(\mathbf{x})}{\partial \theta_{bJ}} \; \frac{\partial e_i(\mathbf{x})}{\partial H_0} \; \frac{\partial e_i(\mathbf{x})}{\partial \tau}\right]^T \tag{61}$$

where

$$\frac{\partial e_i(\mathbf{x})}{\partial r_{ak}} = [2r_{ak} - 2\cos(\theta_{ak})e^{j\omega_i T}]\frac{F(\mathbf{x},\omega_i)}{N_k(e^{j\omega_i T})}$$
$$\frac{\partial e_i(\mathbf{x})}{\partial \theta_{ak}} = [2r_{ak}\sin(\theta_{ak})e^{j\omega_i T}]\frac{F(\mathbf{x},\omega_i)}{N_k(e^{j\omega_i T})}$$
$$\frac{\partial e_i(\mathbf{x})}{\partial r_{bk}} = [-2r_{bk} + 2\cos(\theta_{bk})e^{j\omega_i T}]\frac{F(\mathbf{x},\omega_i)}{D_k(e^{j\omega_i T})}$$
$$\frac{\partial e_i(\mathbf{x})}{\partial \theta_{bk}} = [-2r_{bk}\sin(\theta_{bk})e^{j\omega_i T}]\frac{F(\mathbf{x},\omega_i)}{D_k(e^{j\omega_i T})}$$
$$\frac{\partial e_i(\mathbf{x})}{\partial H_0} = \frac{F(\mathbf{x},\omega_i)}{H_0}$$
$$\frac{\partial e_i(\mathbf{x})}{\partial \tau} = j\omega_i F_0(\omega_i).$$

B. Hessian of the Error Function

The Hessian of the error function, $\nabla^2 e_i(\mathbf{x})$, is given by

$$\nabla^2 e_i(\mathbf{x}) = \begin{bmatrix} \frac{\partial e_i^2(\mathbf{x})}{\partial r_{a1}\partial r_{a1}} & \frac{\partial e_i^2(\mathbf{x})}{\partial r_{a1}\partial \theta_{a1}} & \cdots & \frac{\partial e_i^2(\mathbf{x})}{\partial r_{a1}\partial H_0} & \frac{\partial e_i^2(\mathbf{x})}{\partial r_{a1}\partial \tau} \\ \frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{a1}\partial r_{a1}} & \frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{a1}\partial \theta_{a1}} & \cdots & \frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{a1}\partial H_0} & \frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{a1}\partial \tau} \\ \frac{\partial e_i^2(\mathbf{x})}{\partial r_{b1}\partial r_{a1}} & \frac{\partial e_i^2(\mathbf{x})}{\partial r_{b1}\partial \theta_{a1}} & \cdots & \frac{\partial e_i^2(\mathbf{x})}{\partial r_{b1}\partial H_0} & \frac{\partial e_i^2(\mathbf{x})}{\partial r_{b1}\partial \tau} \\ \frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{b1}\partial r_{a1}} & \frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{b1}\partial \theta_{a1}} & \cdots & \frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{b1}\partial H_0} & \frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{b1}\partial \tau} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{\partial e_i^2(\mathbf{x})}{\partial H_0\partial r_{a1}} & \frac{\partial e_i^2(\mathbf{x})}{\partial H_0\partial \theta_{a1}} & \cdots & \frac{\partial e_i^2(\mathbf{x})}{\partial H_0\partial H_0} & \frac{\partial e_i^2(\mathbf{x})}{\partial H_0\partial \tau} \\ \frac{\partial e_i^2(\mathbf{x})}{\partial \tau\partial r_{a1}} & \frac{\partial e_i^2(\mathbf{x})}{\partial \tau\partial \theta_{a1}} & \cdots & \frac{\partial e_i^2(\mathbf{x})}{\partial \tau\partial H_0} & \frac{\partial e_i^2(\mathbf{x})}{\partial \tau\partial \tau} \end{bmatrix}. \tag{62}$$

As for the polynomial formulation in Appendix A, the Hessian in (62) is not block diagonal and several equations are required to construct it. Its construction can be simplified by splitting it into several block symmetric matrices and taking advantage of its symmetry properties. The second-order partial derivatives with respect to the zero/pole radii and angles are divided into two block matrix types, diagonal 4 × 4 block matrices and lower triangular 4 × 4 block matrices.

The diagonal block matrices are given by

$$H_{mm}^{(\mathrm{diag})} = \begin{bmatrix} \frac{\partial e_i^2(\mathbf{x})}{\partial r_{am}\partial r_{am}} & \frac{\partial e_i^2(\mathbf{x})}{\partial r_{am}\partial \theta_{am}} & \frac{\partial e_i^2(\mathbf{x})}{\partial r_{am}\partial r_{bm}} & \frac{\partial e_i^2(\mathbf{x})}{\partial r_{am}\partial \theta_{bm}} \\ \frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{am}\partial r_{am}} & \frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{am}\partial \theta_{am}} & \frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{am}\partial r_{bm}} & \frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{am}\partial \theta_{bm}} \\ \frac{\partial e_i^2(\mathbf{x})}{\partial r_{bm}\partial r_{am}} & \frac{\partial e_i^2(\mathbf{x})}{\partial r_{bm}\partial \theta_{am}} & \frac{\partial e_i^2(\mathbf{x})}{\partial r_{bm}\partial r_{bm}} & \frac{\partial e_i^2(\mathbf{x})}{\partial r_{bm}\partial \theta_{bm}} \\ \frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{bm}\partial r_{am}} & \frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{bm}\partial \theta_{am}} & \frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{bm}\partial r_{bm}} & \frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{bm}\partial \theta_{bm}} \end{bmatrix} \tag{63}$$

where $m = 1, \ldots, J$ is the number of sections. The $H_{mm}^{(\mathrm{diag})}$ block matrices are symmetric blocks that are located along the diagonal of the Hessian and represent all the second-order partial derivative combinations with respect to the zero/pole radii and angles for the same biquadratic factor. From the symmetry of the Hessian, the upper triangular matrix is the reflection of the lower triangular matrix and, therefore, only the equations for the lower triangular matrix and the diagonal second-order partial derivatives are required.

Another interesting observation is that, unlike the Hessian diagonal blocks for the polynomial formulation, the upper 2 × 2 diagonal block submatrices of the $H_{mm}^{(\mathrm{diag})}$ blocks are not equal to zero. The equations for the $H_{mm}^{(\mathrm{diag})}$ block matrices follow. If we let

$$q(\theta) = \cos(\theta)e^{j\omega_i T} \quad \text{and} \quad s(\theta) = \sin(\theta)e^{j\omega_i T}$$

we get

$$\frac{\partial e_i^2(\mathbf{x})}{\partial r_{am}\partial r_{am}} = \frac{2F(\mathbf{x},\omega_i)}{N_m(e^{j\omega_i T})}$$
$$\frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{am}\partial r_{am}} = \frac{2s(\theta_{am})F(\mathbf{x},\omega_i)}{N_m(e^{j\omega_i T})}$$
$$\frac{\partial e_i^2(\mathbf{x})}{\partial r_{bm}\partial r_{am}} = \frac{[2r_{am} - 2q(\theta_{am})][-2r_{bm} + 2q(\theta_{bm})]F(\mathbf{x},\omega_i)}{N_m(e^{j\omega_i T})D_m(e^{j\omega_i T})}$$
$$\frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{bm}\partial r_{am}} = \frac{[2r_{am} - 2q(\theta_{am})][-2r_{bm}s(\theta_{bm})]F(\mathbf{x},\omega_i)}{N_m(e^{j\omega_i T})D_m(e^{j\omega_i T})}$$
$$\frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{am}\partial \theta_{am}} = \frac{2r_{am}q(\theta_{am})F(\mathbf{x},\omega_i)}{N_m(e^{j\omega_i T})}$$

$$\begin{aligned}
\frac{\partial e_i^2(\mathbf{x})}{\partial r_{bm}\partial \theta_{am}} &= \frac{2r_{am}s(\theta_{am})[-2r_{bm}+2q(\theta_{bm})]F(\mathbf{x},\omega_i)}{N_m(e^{j\omega_i T})D_m(e^{j\omega_i T})}\\
\frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{bm}\partial \theta_{am}} &= \frac{2r_{am}s(\theta_{am})\left[-2r_{bm}s(\theta_{bm})\right]F(\mathbf{x},\omega_i)}{N_m(e^{j\omega_i T})D_m(e^{j\omega_i T})}\\
\frac{\partial e_i^2(\mathbf{x})}{\partial r_{bm}\partial r_{bm}} &= \frac{-2F(\mathbf{x},\omega_i)}{D_m(e^{j\omega_i T})}\\
&\quad + \frac{[-2r_{bm}+2q(\theta_{bm})][-4r_{bm}+4q(\theta_{bm})]F(\mathbf{x},\omega_i)}{D_m^2\left(e^{j\omega_i T}\right)}\\
\frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{bm}\partial r_{bm}} &= \frac{F(\mathbf{x},\omega_i)}{D_m(e^{j\omega_i T})}\left(-2s(\theta_{bm})\right.\\
&\quad \left. + \frac{[-2r_{bm}+2q(\theta_{bm})][-4r_{bm}s(\theta_{bm})]}{D_m(e^{j\omega_i T})}\right)\\
\frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{bm}\partial \theta_{bm}} &= \frac{F(\mathbf{x},\omega_i)}{D_m(e^{j\omega_i T})}\left(-2q(\theta_{bm})\right.\\
&\quad \left. + \frac{[-2r_{bm}s(\theta_{bm})][-4r_{bm}s(\theta_{bm})]}{D_m(e^{j\omega_i T})}\right)
\end{aligned}$$

The lower triangular blocks are given by

$$H_{jm}^{(\text{lower})} = \begin{bmatrix}
\frac{\partial e_i^2(\mathbf{x})}{\partial r_{aj}\partial r_{am}} & \frac{\partial e_i^2(\mathbf{x})}{\partial r_{aj}\partial \theta_{am}} & \frac{\partial e_i^2(\mathbf{x})}{\partial r_{aj}\partial r_{bm}} & \frac{\partial e_i^2(\mathbf{x})}{\partial r_{aj}\partial \theta_{bm}}\\
\frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{aj}\partial r_{am}} & \frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{aj}\partial \theta_{am}} & \frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{aj}\partial r_{bm}} & \frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{aj}\partial \theta_{bm}}\\
\frac{\partial e_i^2(\mathbf{x})}{\partial r_{bj}\partial r_{am}} & \frac{\partial e_i^2(\mathbf{x})}{\partial r_{bj}\partial \theta_{am}} & \frac{\partial e_i^2(\mathbf{x})}{\partial r_{bj}\partial r_{bm}} & \frac{\partial e_i^2(\mathbf{x})}{\partial r_{bj}\partial \theta_{bm}}\\
\frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{bj}\partial r_{am}} & \frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{bj}\partial \theta_{am}} & \frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{bj}\partial r_{bm}} & \frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{bj}\partial \theta_{bm}}
\end{bmatrix} \tag{64}$$

where $j = 1,\ldots,J, m = 1,\ldots,J, j \neq m$. Unlike the diagonal block matrices, the lower-triangular block matrices are not symmetric and, therefore, there are 16 equations for each block. The $H_{jm}^{(\text{lower})}$ block matrix elements represent the second-order partial derivative combinations with respect to the transfer function coefficients for the cases where $j \neq m$. The equations for the first four columns of the $H_{jm}^{(\text{lower})}$ block matrices are given below.

First column:

$$\begin{aligned}
\frac{\partial e_i^2(\mathbf{x})}{\partial r_{aj}\partial r_{am}} &= \frac{[2r_{ram}-2q(\theta_{am})][2r_{raj}-2q(\theta_{aj})]F(\mathbf{x},\omega_i)}{N_j(e^{j\omega_i T})N_m\left(e^{j\omega_i T}\right)}\\
\frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{aj}\partial r_{am}} &= \frac{[2r_{ram}-2q(\theta_{am})][2r_{raj}s(\theta_{aj})]F(\mathbf{x},\omega_i)}{N_j\left(e^{j\omega_i T}\right)N_m\left(e^{j\omega_i T}\right)}\\
\frac{\partial e_i^2(\mathbf{x})}{\partial r_{bm}\partial r_{am}} &= \frac{[2r_{am}-2q(\theta_{am})][-2r_{bj}+2q(\theta_{bj})]F(\mathbf{x},\omega_i)}{N_m\left(e^{j\omega_i T}\right)D_j\left(e^{j\omega_i T}\right)}\\
\frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{bj}\partial r_{am}} &= \frac{[2r_{am}-2q(\theta_{am})][-2r_{bj}s(\theta_{bj})]F(\mathbf{x},\omega_i)}{N_m\left(e^{j\omega_i T}\right)D_j\left(e^{j\omega_i T}\right)}
\end{aligned}$$

Second column:

$$\begin{aligned}
\frac{\partial e_i^2(\mathbf{x})}{\partial r_{aj}\partial \theta_{am}} &= \frac{2r_{am}s(\theta_{am})[2r_{aj}-2q(\theta_{aj})]F(\mathbf{x},\omega_i)}{N_j\left(e^{j\omega_i T}\right)N_m\left(e^{j\omega_i T}\right)}\\
\frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{aj}\partial \theta_{am}} &= \frac{2r_{am}s(\theta_{am})[2r_{aj}s(\theta_{aj})]F(\mathbf{x},\omega_i)}{N_j\left(e^{j\omega_i T}\right)N_m\left(e^{j\omega_i T}\right)}\\
\frac{\partial e_i^2(\mathbf{x})}{\partial r_{bj}\partial \theta_{am}} &= \frac{2r_{am}s(\theta_{am})[-2r_{bj}+2q(\theta_{bj})]F(\mathbf{x},\omega_i)}{N_m\left(e^{j\omega_i T}\right)D_j\left(e^{j\omega_i T}\right)}\\
\frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{bj}\partial \theta_{am}} &= \frac{2r_{am}s(\theta_{am})[-2r_{bj}s(\theta_{bj})]F(\mathbf{x},\omega_i)}{N_m\left(e^{j\omega_i T}\right)D_j\left(e^{j\omega_i T}\right)}
\end{aligned}$$

Third column:

$$\begin{aligned}
\frac{\partial e_i^2(\mathbf{x})}{\partial r_{aj}\partial r_{bm}} &= \frac{[-2r_{bm}+2q(\theta_{bm})][2r_{aj}-2q(\theta_{aj})]F(\mathbf{x},\omega_i)}{N_j\left(e^{j\omega_i T}\right)N_m\left(e^{j\omega_i T}\right)}\\
\frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{aj}\partial r_{bm}} &= \frac{[-2r_{bm}+2q(\theta_{bm})][2r_{aj}s(\theta_{aj})]F(\mathbf{x},\omega_i)}{N_j\left(e^{j\omega_i T}\right)N_m\left(e^{j\omega_i T}\right)}\\
\frac{\partial e_i^2(\mathbf{x})}{\partial r_{bj}\partial r_{bm}} &= \frac{[-2r_{bm}+2q(\theta_{bm})][-2r_{bj}+2q(\theta_{bj})]F(\mathbf{x},\omega_i)}{N_m\left(e^{j\omega_i T}\right)D_j\left(e^{j\omega_i T}\right)}\\
\frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{bj}\partial r_{bm}} &= \frac{[-2r_{bm}+2q(\theta_{bm})][-2r_{bj}s(\theta_{bj})]F(\mathbf{x},\omega_i)}{N_m\left(e^{j\omega_i T}\right)D_j\left(e^{j\omega_i T}\right)}
\end{aligned}$$

Fourth column:

$$\begin{aligned}
\frac{\partial e_i^2(\mathbf{x})}{\partial r_{aj}\partial \theta_{bm}} &= \frac{-2r_{bm}s(\theta_{bm})[2r_{aj}-2q(\theta_{aj})]F(\mathbf{x},\omega_i)}{N_j\left(e^{j\omega_i T}\right)N_m\left(e^{j\omega_i T}\right)}\\
\frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{aj}\partial \theta_{bm}} &= \frac{-2r_{bm}s(\theta_{bm})[2r_{aj}s(\theta_{aj})]F(\mathbf{x},\omega_i)}{N_j\left(e^{j\omega_i T}\right)N_m\left(e^{j\omega_i T}\right)}\\
\frac{\partial e_i^2(\mathbf{x})}{\partial r_{bj}\partial \theta_{bm}} &= \frac{-2r_{bm}s(\theta_{bm})[-2r_{bj}+2q(\theta_{bj})]F(\mathbf{x},\omega_i)}{N_m\left(e^{j\omega_i T}\right)D_j\left(e^{j\omega_i T}\right)}\\
\frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{bj}\partial \theta_{bm}} &= \frac{-2r_{bm}s(\theta_{bm})[-2r_{bj}s(\theta_{bj})]F(\mathbf{x},\omega_i)}{N_m\left(e^{j\omega_i T}\right)D_j\left(e^{j\omega_i T}\right)}
\end{aligned}$$

The second-order partial derivatives with respect to the transfer function multiplier constant, H_0, are

$$\begin{aligned}
\frac{\partial e_i^2(\mathbf{x})}{\partial H_0\partial r_{am}} &= \frac{[2r_{am}-2q(\theta_{am})]F(\mathbf{x},\omega_i)}{H_0N_m\left(e^{j\omega_i T}\right)}\\
\frac{\partial e_i^2(\mathbf{x})}{\partial H_0\partial \theta_{am}} &= \frac{[2r_{am}s(\theta_{am})]F(\mathbf{x},\omega_i)}{H_0N_m\left(e^{j\omega_i T}\right)}\\
\frac{\partial e_i^2(\mathbf{x})}{\partial H_0\partial r_{bm}} &= \frac{[-2r_{bm}+2q(\theta_{bm})]F(\mathbf{x},\omega_i)}{H_0D_m\left(e^{j\omega_i T}\right)}\\
\frac{\partial e_i^2(\mathbf{x})}{\partial H_0\partial \theta_{bm}} &= \frac{[-2r_{bm}2s(\theta_{am})]F(\mathbf{x},\omega_i)}{H_0D_m(e^{j\omega_i T})}
\end{aligned}$$

Finally, the second-order partial derivatives with respect to the group delay, τ, are

$$\begin{aligned}
&\frac{\partial e_i^2(\mathbf{x})}{\partial \tau\partial r_{am}} = \frac{\partial e_i^2(\mathbf{x})}{\partial \tau\partial \theta_{am}} = \frac{\partial e_i^2(\mathbf{x})}{\partial \tau\partial r_{bm}} = \frac{\partial e_i^2(\mathbf{x})}{\partial \tau\partial \theta_{bm}} = 0\\
&\frac{\partial e_i^2(\mathbf{x})}{\partial H_0\partial H_0} = 0, \quad \frac{\partial e_i^2(\mathbf{x})}{\partial \tau\partial H_0} = 0,\\
\text{and}\quad &\frac{\partial e_i^2(\mathbf{x})}{\partial \tau\partial \tau} = -\,\omega_i^2 M_0(\omega_i)e^{-j\omega_i\tau}.
\end{aligned}$$

References

[1] A. Antoniou, *Digital Signal Processing: Signals, Systems, and Filters*. New York: McGraw-Hill, 2005.

[2] D. J. Shpak, "MATLAB® functions iirlpnorm and iirlpnormc," MATLAB® Filter Design Toolbox 2002, The MathWorks.

[3] C. Y.-F. Ho, B. W.-K. Ling, Y.-Q. Liu, P. K.-S. Tam, and K.-L. Teo, "Optimal design of magnitude responses of rational infinite impulse response filters," *IEEE Trans. Signal Process.*, vol. 54, no. 10, pp. 4039–4046, Oct. 2006.

[4] A. G. Deczky, "Equiripple and minimax (Chebyshev) approximation for recursive digital filters," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-23, pp. 98–111, Aug. 1974.

[5] C. Charalambous and A. Antoniou, "Equalisation of recursive digital filters," *Proc. Inst. Elect. Eng.*, vol. 127, no. 5, pt. G, pp. 219–225, Oct. 1980.

[6] N. Ko, D. J. Shpak, and A. Antoniou, "Design of recursive delay equalizers using constrained optimization," in *IEEE Pacific Rim Conf. Comm., Comp. and Signal Process.*, Victoria, Canada, Aug. 1997, pp. 173–177.

[7] D. J. Shpak, "MATLAB® function iirgrpdelay," MATLAB® Filter Design Toolbox 2002, The MathWorks.

[8] T. Inukai, "A unified approach to optimal recursive digital filter design," *IEEE Trans. Circuits Syst.*, vol. CAS-27, no. 7, pp. 646–649, Jul. 1980.

[9] M. C. Lang, "Least-squares design of IIR filters with prescribed magnitude and phase responses and a pole radius constraint," *IEEE Trans. Signal Process.*, vol. 48, no. 11, pp. 3109–3121, Nov. 2000.

[10] A. Tarczyński, G. D. Cain, E. Hermanowicz, and M. Rojewski, "A WISE method for designing IIR filters," *IEEE Trans. Signal Process.*, vol. 49, no. 7, pp. 1421–1432, Jul. 2001.

[11] B. Dumitrescu and R. Niemist, "Multistage IIR filter design using convex stability domains defined by positive realness," *IEEE Trans. Signal Process.*, vol. 52, no. 4, pp. 962–974, Apr. 2004.

[12] C.-C. Tseng, "Design of stable IIR digital filter based on least p-power error criterion," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 51, no. 9, pp. 1879–1888, Sep. 2004.

[13] W.-S. Lu, "Digital filter design: Global solutions via polynomial optimization," in *Proc. IEEE Asian Pacific Conf. Circuits Syst.*, Dec. 2006, pp. 49–52.

[14] W.-S. Lu, "An argument-principle based stability criterion and application to the design of IIR digital filters," in *Proc. Intern. Symp. Circuits Syst.*, 2006, pp. 4431–4434.

[15] S. C. Chan, K. M. Tsui, and K. W. Tse, "Design of constrained causal stable IIR filters using a new second-order-cone-programming-based model-reduction technique," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 54, no. 2, pp. 107–111, Feb. 2007.

[16] N. Ko, "Design of Recursive Delay Equalizers by Constrained Optimization," M.Sc. thesis, Dep. Elect. Comput. Eng., Univ. of Victoria, Victoria, Canada, 2001.

[17] A. Antoniou and W.-S. Lu, *Practical Optimization: Algorithms and Engineering Applications*. New York: Springer, 2007.

[18] R. Fletcher, *Practical Methods of Optimization*, 2nd ed. New York: Wiley, 1993.

[19] A. Antoniou, "Improved minimax optimization algorithms and their application in the design of recursive digital filters," *IEE Proc.*, vol. 138, no. 6, pt. G, pp. 724–730, Dec. 1991.

[20] C. Charalambous, "A unified review of optimization," *IEEE Trans. Microw. Theory Techn.*, vol. MTT-22, pp. 289–300, Mar. 1974.

[21] W.-S. Lu, "Design of recursive digital filters with prescribed stability margin: A parameterization approach," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 45, pp. 1289–1298, Sep. 1998.

[22] S. Saab, A. Antoniou, and W.-S. Lu, "Design of linear-phase recursive digital filters by optimization," in *Proc. Int. Symp. on Adv. in Digital Filtering and Signal Process.*, Jun. 1998, pp. 87–91.

[23] I. Kale, J. Gryka, G. D. Cain, and B. Beliczynski, "FIR filter order reduction: Balanced model truncation and Hankel-norm optimal approximation," *IEEE Proc. Vis. Image Signal Process.*, vol. 141, pp. 168–174, Jun. 1994.

[24] B. Beliczynski, J. Gryka, and I. Kale, "Critical comparison of Hankel-norm optimal approximation and balanced model truncation algorithms as vehicles for FIR-to-IIR filter order reduction," in *IEEE Inter. Conf. Acoust., Speech, and Signal Process.*, Apr. 1994, vol. 3, pp. 593–596.

[25] B. C. Moore, "Principal component analysis in linear systems: Controllability, observability and model reduction," *IEEE Trans. Autom. Control*, vol. AC-26, no. 2, pp. 17–31, Feb. 1981.

[26] L. Prenobo and L. M. Silverman, "Model reduction via balanced state space representation," *IEEE Trans. Autom. Control*, vol. AC-27, no. 4, pp. 382–387, Apr. 1982.

[27] W.-S. Lu and T. Hinamoto, "Optimal design of IIR digital filters with robust stability using conic-quadratic-programming updates," *IEEE Trans. Signal Process.*, vol. 51, no. 6, pp. 1581–1592, Jun. 2003.

[28] A. G. Deczky, "Synthesis of recursive digital filters using the minimum p-error criterion," *IEEE Trans. Audio Electroacoust.*, vol. AU-20, no. 5, pp. 257–263, Oct. 1972.

[29] W.-S. Lu, "Design of stable IIR digital filters with equiripple passbands and peak-constrained least squares stopbands," in *Proc. Intern. Symp. Circuits Syst.*, 1997, pp. 2192–2195.

David Guindon received the B.Eng and M.A.Sc. degrees from the University of Victoria, Victoria, BC, Canada, in 2001 and 2008, respectively.

His current interests include computer programming, innovative software, graphics design, and online marketing.

Dale J. Shpak (S'79–M'86–SM'09) received the B.Sc.(Elec. Eng.) degree in electrical engineering from the University of Calgary, Canada in 1980, and the M.Eng. degree in electronics in 1992, while working as an engineer for the City of Calgary Electric System.

He then worked as a researcher in computer systems, electronics, and DSP between 1982 and 1987. From 1987 to 1989, he earned his Ph.D. at the University of Victoria, Canada. He joined the Department of Engineering at Royal Roads Military College in 1988, remaining there until it closed in 1995. Since 1989 he has held an adjunct position at the University of Victoria. As an Adjunct Professor of Electrical and Computer Engineering he receives NSERC funding for ongoing research programs with his graduate students. He joined the Department of Computer Science at Camosun College in 1999, where he lectures on subjects including object-oriented programming, computer networks, software engineering, and real-time and concurrent systems. He has held several positions in industry where he developed software, circuits, networking systems, and embedded systems. He is a principal developer of award-winning products including the Filter Design Toolbox for MATLAB™. He currently develops software and embedded systems for products ranging from audio processing to wireless control systems. His principal research interests are in the areas of signal processing for communications and music, design of software systems, embedded systems design, and digital filter design.

Dr. Shpak is a member of the Association of Professional Engineers of the Province of British Columbia.

Andreas Antoniou (M'69–SM'79–F'82–LF'04) received the B.Sc.(Eng.) and Ph.D. degrees in electrical engineering from the University of London, London, U.K., in 1963 and 1966, respectively,

He taught at Concordia University from 1970 to 1983, was the founding Chair of the Department of Electrical and Computer Engineering, University of Victoria, B.C., Canada, from 1983 to 1990, and is now Professor Emeritus. His teaching and research interests are in the area of digital signal processing. He is the author of *Digital Signal Processing: Signals, Systems, and Filters* (McGraw-Hill, 2005) and the co-author with Wu-Sheng Lu of *Practical Optimization: Algorithms and Engineering Applications* (Springer, 2007).

Dr. Antoniou served as Associate/Chief Editor for IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS (CAS) from 1983 to 1987, as a Distinguished Lecturer of the IEEE Signal Processing and the Circuits and Systems Societies during 2003–2004 and 2006–2007, respectively, and as General Chair of the 2004 International Symposium on Circuits and Systems. He is a Fellow of the IET, U.K. He was awarded the CAS Golden Jubilee Medal by the IEEE Circuits and Systems Society, the B.C. Science Council Chairmans Award for Career Achievement for 2000, the Doctor Honoris Causa degree by the National Technical University, Athens, Greece, in 2002, the IEEE Circuits and Systems Society Technical Achievement Award for 2005, the 2008 IEEE Canada Outstanding Engineering Educator Silver Medal, and the IEEE Circuits and Systems Society Education Award for 2009.