Design of Recursive Digital Filters with Prescribed Stability Margin: A Parameterization Approach

Wu-Sheng Lu, *Senior Member, IEEE*

***Abstract*—A major problem in the design of recursive digital filters is stability. Although an unstable recursive filter obtained from a design algorithm can be stabilized by reciprocal substitution of the unstable factors of the filter without changing its amplitude response, this stabilization technique cannot be applied if phase response is a part of the design specifications. In this paper, we propose a new method for the design of recursive digital filters with a prescribed stability margin by parameterizing all stable transfer functions and carrying out unconstrained optimization over this class of transfer functions. Three parameterization techniques are described, and closed-form formulas for the gradient functions and Hessian matrices of several typical objective functions are derived. The design technique is expected to be useful in the cases where both amplitude and phase specifications are required in the design.**

***Index Terms*—Optimization, parameterization, recursive digital filters.**

I. Introduction

RECURSIVE digital filters have been extensively used in a variety of applications where high selectivity and efficient processing of discrete signals are required [1]–[6]. A major problem in designing recursive filters is stability. Some design methods deal with this problem by choosing an initial point which corresponds to a stable transfer function, and monitoring the stability of the transfer function generated in each iteration; other design methods do not verify the stability of the intermediate transfer functions, but carry out a stabilization step by reciprocal substitution for the unstable factors in the filter's denominator if the final transfer function turns out to be unstable. However, this stabilization technique cannot be used if phase response is a part of the design specifications since the reciprocal substitution changes the phase response of the filter.

As an approximation problem, the design at hand can naturally be tackled using optimization techniques. In [7], a linear-programming technique was proposed for this purpose. Stability of the infinite-impulse-response (IIR) filters obtained using this technique are assured by imposing a set of *linear* conditions on the real part of the denominate polynomial at dense grid points from $\omega = 0$ to $\omega = \omega_s/2$. Although the linearity of these constraints makes it possible to utilize efficient linear-programming methods, they are *sufficient* (but not necessary) stability conditions, meaning that many stable designs are excluded when a linear programming method is used to search an "optimal" solution. Consequently, the outcome of this method is actually a suboptimal design. In [8], the design problem was addressed using an efficient constrained nonlinear optimization method known as quadratic programming [9]–[11]. However, just as in the case of [7], the linear stability constraints are sufficient (but not necessary), inevitably leading to a certain degree of performance degradation. Although, in principle, other constrained nonlinear optimization methods such as gradient projection methods, reduced gradient methods [9]–[11], and recently developed convex programming methods [12], [13] have the potential of becoming effective problem solvers, "the study of constrained optimization is by no means as well advanced as for the unconstrained case," and "the writing of software is a much more complex task" [11, p. 139]. Another approach that several researchers have taken is to extend the concept of eigenfilters to the IIR case [14]–[16]. However, stability remains an unsolved issue in this class of methods [16].

In this paper, we propose a new method for the design of recursive digital filters with a prescribed stability margin by parameterizing all stable transfer functions and then carrying out *unconstrained* optimization over this class of transfer functions. We propose three parameterization techniques. The first utilizes hyperbolic tangent transformations (HTT's) to characterize all discrete-time, first-, and second-order transfer functions with a prescribed stability margin. The second technique is similar to the first in spirit except that it uses an arc–tangent transformation (ATT). The third approach applies a modified bilinear transfer function to map the stable first- and second-order continuous-time transfer functions to their discrete counterpart with a prescribed stability margin. In these methods, the number of parameters in the denominator remains equal to the order of the denominator polynomial, and the parameters can vary over the *entire* parameter space without violating the stability of the transfer function. Furthermore, we derive closed-form formulas for evaluating the gradient and Hessian matrix of least pth- and minimax-type objective functions. An immediate advantage of the proposed method is that the design can be accomplished using unconstrained optimization techniques [9]–[11], and a prescribed stability margin of the filter is guaranteed. The proposed method is

Manuscript received October 1, 1996; revised April 8, 1998. This work was supported by the Natural Science and Engineering Research Council of Canada. This paper was recommended by Associate Editor B. A. Shenoi.

The author is with the Department of Electrical and Computer Engineering, University of Victoria, Victoria, BC V8W 3P6, Canada.

Publisher Item Identifier S 1057-7130(98)06710-X.

expected to be of use for the cases where both amplitude and phase specifications are required in the design [17]. This paper is organized as follows. Two methods of parameterizing Schur polynomials with a prescribed stability margin using hyperbolic tangent and ATT's are described in Section II. Section III proposes another parameterization method based on the MBT. In Section IV, the problem of designing stable recursive digital filters is formulated as unconstrained optimization problems, and closed-form formulas for evaluating the gradient functions and Hessian matrices of typical objective functions are derived. In Section V, we present several design examples to illustrate the proposed techniques.

II. Parameterization of Schur Polynomials with Prescribed Stability Margin By HTT's T and ATT's

A monic polynomial $p(z)$ with real coefficients is said to be a Schur polynomial if the roots of $p(z) = 0$ are located strictly inside the unit circle. It is well known that a digital filter is stable if and only if the denominator of its transfer function is a Schur polynomial. In what follows, we describe two methods for parameterizing all first- and second-order Schur polynomials.

A. Parameterization Using HTT

The hyperbolic tangent function is defined by

$$u(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}. \tag{1}$$

It is a differentiable monotonically increasing function that maps the entire one-dimensional (1-D) space $-\infty < x < \infty$ to the open interval $-1 < u < 1$. It follows from (1) that

$$p_1(z, b_0) = z + \tanh(b_0), \qquad -\infty < b_0 < \infty \tag{2}$$

characterizes *all* first-order Schur polynomials. Note that only one parameter—b_0—is involved in (2), and it can take any real value over the entire 1-D parameter space.

Now consider the second-order monic polynomial

$$p(z, d_0, d_1) = z^2 + d_1 z + d_0 \tag{3}$$

where d_0 and d_1 are real parameters. It is known [1], [18] that $p(z, d_0, d_1)$ is a Schur polynomial if and only if

$$d_0 < 1 \tag{4a}$$

$$d_1 - d_0 < 1 \tag{4b}$$

$$d_1 + d_0 > -1. \tag{4c}$$

Hence, the stability region for $p(z, d_0, d_1)$ is an open triangle in the parameter space depicted in Fig. 1.

If we let

$$d_0 = \tanh(b_0), \qquad -\infty < b_0 < \infty \tag{5}$$

then d_0 varies over the interval $-1 < d_0 < 1$ when parameter b_0 varies from $-\infty$ to ∞. Further, for a fixed d_0 with $-1 < d_0 < 1$, the value of d_1 may vary from $-(1 + d_0)$ to $1 + d_0$ in order for the point (d_0, d_1) to be inside the stability region. Hence, if we let

$$d_1 = [1 + \tanh(b_0)]\tanh(b_1), \qquad -\infty < b_0, b_1 < \infty \tag{6}$$

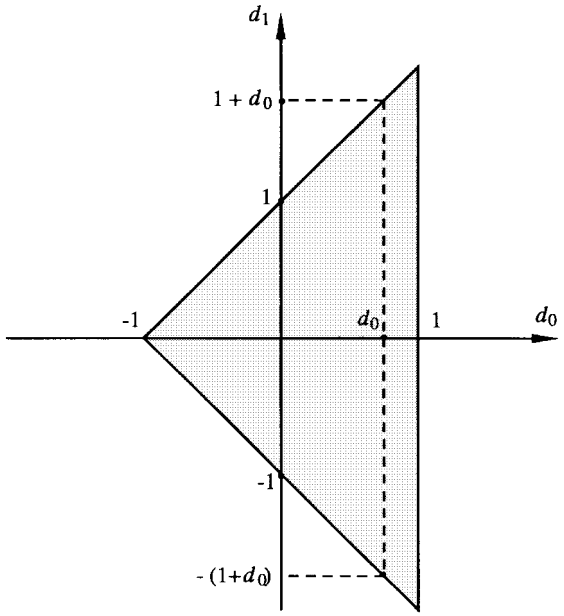


Fig. 1. Stability region of polynomial $p(z, d_0, d_1)$.

then (5) and (6) jointly define a transformation which provides a one-to-one mapping between the entire parameter space (b_0, b_1) and the stability region in parameter space (d_0, d_1). In other words,

$$p_2(z, b_0, b_1) = z^2 + [1 + \tanh(b_0)]\tanh(b_1)z + \tanh(b_0) \tag{7}$$

with $-\infty < b_0, b_1 < \infty$ characterizing *all* second-order Schur polynomials where two real parameters b_0 and b_1 can take any values in the entire two-dimensional (2-D) parameter space.

B. Parameterization Using ATT

An alternative approach to the parameterization of all first- and second-order Schur polynomials is to use the ATT

$$v(x) = \frac{2}{\pi}\tan^{-1}(x). \tag{8}$$

Note that $v(x)$ is a differentiable and monotonically increasing function that maps the entire 1-D space into the interval $-1 < v < 1$.

By using an argument similar to that in Section II–A, we can show that

$$q_1(z, b_0) = z + \frac{2}{\pi}\tan^{-1}(b_0), \qquad -\infty < b_0 < \infty \tag{9}$$

characterizes *all* first-order Schur polynomials, and

$$q_2(z, b_0, b_1) = z^2 + \frac{2}{\pi}\left[1 + \frac{2}{\pi}\tan^{-1}(b_0)\right] \cdot \tan^{-1}(b_1)z + \frac{2}{\pi}\tan^{-1}(b_0) \tag{10}$$

with $-\infty > b_0, b_1 < \infty$ characterizing *all* second-order Schur polynomials.

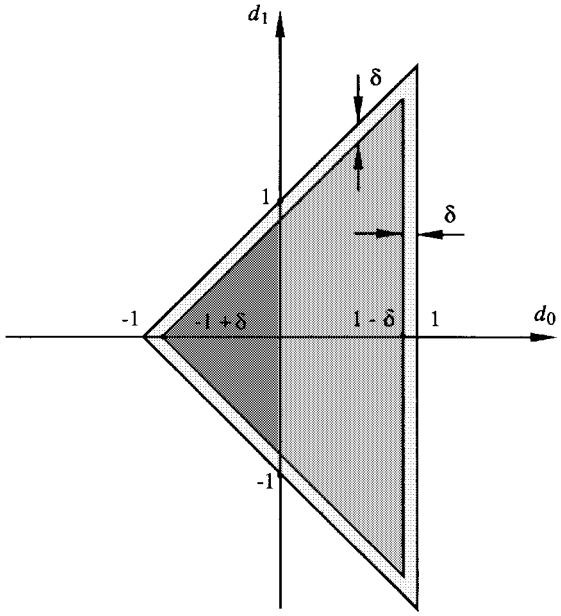


Fig. 2. Stability region with a stability margin δ.

C. Schur Polynomials with Prescribed Stability Margin

When implementing a stable recursive filter, rounding or truncation of the filter coefficients may lead to a unstable implementation if the stability margin of the filter is too small. It is, therefore, desirable to approximate a given frequency response by a transfer function with a prescribed stability margin. This means that the zero of the first-order polynomial factor in the denominator should be within interval $(-1+\delta, 1-\delta)$ with some $\delta > 0$, and that the zeros of all second-order polynomial factors in the denominators should be within the darker region shown in Fig. 2, where $\delta > 0$ defines a kind of stability margin for the polynomial in (3).

Since the function

$$u^{(\delta)}(x) = (1-\delta)\tanh(x)$$

maps the entire 1-D space $-\infty < x < \infty$ to the open interval $(-1+\delta, 1-\delta)$, the HTT can be used to parameterize all first-order Schur polynomials with stability margin δ as

$$p_1^{(\delta)}(z, b_0) = z + (1-\delta)\tanh(b_0), \qquad -\infty < b_0 < \infty \tag{11}$$

and to parameterize all second-order Schur polynomials with stability margin δ as

$$p_2^{(\delta)}(z, b_0, b_1) = z^2 + (1-\delta)[1+\tanh(b_0)]\tanh(b_1)z + (1-\delta)\tanh(b_0) \tag{12}$$

with $-\infty < b_0,\ b_1 < \infty$.

Similarly, the ATT can be used to parameterize all first-order Schur polynomials with stability margin δ as

$$q_1^{(\delta)}(z, b_0) = z + \frac{2(1-\delta)}{\pi}\tan^{-1}(b_0), \qquad -\infty < b_0 < \infty \tag{13}$$

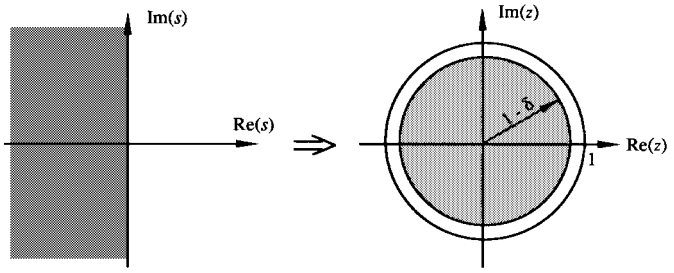


Fig. 3. The MBT maps the left-half s-plane into the disc with radius $1-\delta$.

and to parameterize all second-order Schur polynomials with stability margin δ as

$$q_2^{(\delta)}(z, b_0, b_1) = z^2 + \frac{2(1-\delta)}{\pi}\left[1 + \frac{2}{\pi}\tan^{-1}(b_0)\right] \cdot \tan^{-1}(b_1)z + \frac{2(1-\delta)}{\pi}\tan^{-1}(b_0) \tag{14}$$

with $-\infty < b_0,\ b_1 < \infty$.

III. Parameterization Based on the Modified Bilinear Transformation

A monic real-coefficient polynomial $h(s)$ is said to be a Hurwitz polynomial if the roots of $h(s) = 0$ have negative real parts. It is well known that the bilinear transformation

$$s = \frac{z-1}{z+1}$$

transforms a Hurwitz polynomial to a Schur polynomial. Defining the modified bilinear transformation (MBT) as

$$s = \frac{z-1+\delta}{z+1-\delta}, \qquad 0 < \delta < 1 \tag{15}$$

it follows that the MBT maps the imaginary axis to the circle centered at the origin with radius $1-\delta$. A graphic illustration of (15) is shown in Fig. 3. We see that the MBT transforms a Hurwitz polynomial into a Schur polynomial with stability margin δ.

A. First-Order Schur Polynomials

It is quite obvious that all first-order Hurwitz polynomials can be characterized by $h(s, b_0) = s + b_0^2$ where b_0 is a *nonzero* real parameter.

Using the MBT, we obtain

$$h(s, b_0)\Big|_{s=(z-1+\delta/z+1-\delta)} = \frac{z + (1-\delta)\dfrac{b_0^2-1}{b_0^2+1}}{(z+1-\delta)/(b_0^2+1)}.$$

Hence, the family of polynomials given by

$$r_1^{(\delta)}(z, b_0) = z + (1-\delta)\frac{b_0^2-1}{b_0^2+1} \tag{16}$$

with $-\infty < b_0 < \infty$ characterizing all first-order Schur polynomials with stability margin δ.

B. Second-Order Schur Polynomials

It is well known that all second-order Hurwitz polynomials can be characterized as $h(s,b_0,b_1)$ of the form

$$h(s,b_0,b_1)=s^2+b_0^2s+b_1^2 \tag{17}$$

where b_0 and b_1 are nonzero real parameters. Applying the MBT to (17), we obtain

$$h(s,b_0,b_1)\Big|_{s=(z-1+\delta/z+1-\delta)} = \frac{z^2+\dfrac{2(1-\delta)(b_1^2-1)}{b_1^2+b_0^2+1}\,z+(1-\delta)^2\,\dfrac{b_1^2-b_0^2+1}{b_1^2+b_0^2+1}}{(z+1-\delta)^2/(b_1^2+b_0^2+1)}.$$

Hence, the family of polynomials $r_2^{(\delta)}(z,b_0,b_1)$ given by

$$r_2^{(\delta)}(z,b_0,b_1)=z^2+\frac{2(1-\delta)(b_1^2-1)}{b_1^2+b_0^2+1}\cdot z+(1-\delta)^2\,\frac{b_1^2-b_0^2+1}{b_1^2+b_0^2+1} \tag{18}$$

with $-\infty<b_0,\,b_1<\infty$ characterizing all second-order Schur polynomials with stability margin δ.

IV. Design of Stable Recursive Digital Filters

A. Typical Objective Functions and Their Gradients

Let the transfer function of a $2K$th-order digital filter be expressed as

$$H(z)=H_0\prod_{k=1}^{K}\frac{N_k(z)}{D_k(z)} \tag{19}$$

where

$$N_k(z)=z^2+a_{1k}z+a_{0k} \tag{20}$$

$$D_k(z)=\begin{cases} p_2^{(\delta)}(z,b_{0k},b_{1k}), & \text{for HTT} \quad (21a)\\ q_2^{(\delta)}(z,b_{0k},b_{1k}), & \text{for ATT} \quad (21b)\\ r_2^{(\delta)}(z,b_{0k},b_{1k}), & \text{for MBT} \quad (21c)\end{cases}$$

where $p_2^{(\delta)}$, $q_2^{(\delta)}$, and $r_2^{(\delta)}$ are defined by (12), (14), and (18), respectively.

Let $\Omega=\{\omega_i,1\le i\le M\}$ be the set of frequencies at which the frequency response of the filter is evaluated, and define

$$e_i(\boldsymbol{x})=F(\boldsymbol{x},\omega_i)-F_0(\omega_i) \tag{22}$$

where $F_0(\omega_i)$ is the desired frequency response at ω_i, and

$$F(\boldsymbol{x},\omega_i)=H_0\prod_{k=0}^{K}\frac{N_k(e^{j\omega_iT})}{D_k(e^{j\omega_iT})}. \tag{23}$$

If a linear phase response is required, then

$$F_0(\omega_i)=e^{-j\tau\omega_i}M_0(\omega_i) \tag{24}$$

where $M_0(\omega_i)$ is the desired amplitude at ω_i and τ is the group delay. The parameter vector $\boldsymbol{x}$ collects the coefficients of $H(z)$ and the group delay τ as follows:

$$\boldsymbol{x}=[H_0\quad\tau\quad a_{01}\quad a_{11}\quad\cdots\quad a_{0K}\quad a_{1K}\quad b_{01}\quad b_{11}\quad\cdots\quad b_{0K}\quad b_{1K}]^T. \tag{25}$$

An important quantity required in many unconstrained optimization algorithms such as the quasi-Newton algorithms and the conjugate gradient algorithms [11], [19] is the gradient of the objective function. For the weighted least pth optimization, the objective function is defined by

$$E(\boldsymbol{x})=\sum_{i=1}^{M}w_i|e_i(\boldsymbol{x})|^p \tag{26}$$

where $p>0$ is an integer, and $\{w_i,1\le i\le M\}$ are weights at frequencies Ω. The gradient of $E(\boldsymbol{x})$ can be evaluated as follows:

$$\begin{aligned}\nabla E(\boldsymbol{x})&=\frac{p}{2}\sum_{i=1}^{M}w_i|e_i(\boldsymbol{x})|^{p-2}\left[\overline{e}_i(\boldsymbol{x})\,\frac{\partial e_i(\boldsymbol{x})}{\partial\boldsymbol{x}}+e_i(\boldsymbol{x})\,\frac{\partial\overline{e}_i(\boldsymbol{x})}{\partial\boldsymbol{x}}\right]\\&=p\sum_{i=1}^{M}w_i|e_i(\boldsymbol{x})|^{p-2}\mathrm{Re}\left[\overline{e}_i(\boldsymbol{x})\,\frac{\partial e_i(\boldsymbol{x})}{\partial\boldsymbol{x}}\right]\end{aligned} \tag{27}$$

where $\overline{e}_i(\boldsymbol{x})$ denotes the complex conjugate of $e_i(\boldsymbol{x})$, and $\partial e_i(\boldsymbol{x})/\partial\boldsymbol{x}$ is given by (28)–(33c), shown at the bottom of the following page, where $\cosh(b_{0k})$ denotes the hyperbolic cosine of b_{0k} defined by $\cosh(b_{0k})=(e^{b_{0k}}+e^{-b_{0k}})/2$.

For the case where the transfer function has odd order $2K+1$, we write the transfer function as

$$H(z)=H_0\left[\prod_{k=1}^{K}\frac{N_k(z)}{D_k(z)}\right]\left[\frac{N_{K+1}(z)}{D_{K+1}(z)}\right] \tag{34}$$

where $N_k(z)$ and $D_k(z)$ for $1\le k\le K$ are defined by (20), (21), and

$$N_{k+1}(z)=z+a_{0,K+1}$$

$$D_{K+1}(z)=\begin{cases} z+(1-\delta)\tanh(b_{0,K+1}), & \text{for HTT}\\ z+\dfrac{2(1-\delta)}{\pi}\tan^{-1}(b_{0,K+1}), & \text{for ATT}\\ z+(1-\delta)\,\dfrac{b_{0,K+1}^2-1}{b_{0,K+1}^2+1}, & \text{for MBT.}\end{cases}$$

It follows that

$$\frac{\partial e_i(\boldsymbol{x})}{\partial a_{0,K+1}}=\frac{F(\boldsymbol{x},\omega_i)}{N_{K+1}(e^{j\omega_iT})} \tag{35}$$

along with (36a)–(36c), shown at the bottom of the following page.

As an alternative, recursive digital filters can be designed using efficient minimax algorithms developed in [20]–[22], where the objective function is defined by

$$\Psi(\boldsymbol{x},\lambda,\xi)=\frac{1}{2}\sum_{i\in I_1}\lambda_i\phi_i^2(\boldsymbol{x},\xi)+\frac{1}{2}\sum_{i\in I_2}\phi_i^2(\boldsymbol{x},\xi) \tag{37}$$

where ξ and λ_i for $i=1,2,\cdots,M$ are constants,

$$\phi_i(\boldsymbol{x},\xi)=|e_i(\boldsymbol{x})|-\xi \tag{38}$$

$$I_1=\{i\colon\phi_i(\boldsymbol{x},\xi)>0\quad\text{and}\quad\lambda_i>0\} \tag{39}$$

and

$$I_2=\{i\colon\phi_i(\boldsymbol{x},\xi)>0\quad\text{and}\quad\lambda_i=0\}. \tag{40}$$

In this case, the gradient of $\Psi(\boldsymbol{x},\lambda,\xi)$ is given by

$$\nabla\Psi(\boldsymbol{x},\lambda_k,\xi) = \sum_{i\in I_1}\lambda_{ki}\phi_i(\boldsymbol{x},\xi)\nabla|e_i(\boldsymbol{x})| + \sum_{i\in I_2}\phi_i(\boldsymbol{x},\xi)\nabla|e_i(\boldsymbol{x})| \quad (41)$$

where

$$\nabla|e_i(\boldsymbol{x})| = \nabla[\overline{e}_i(\boldsymbol{x})e_i(\boldsymbol{x})]^{1/2} = \mathrm{Re}\left[\frac{\overline{e}_i(\boldsymbol{x})}{|e_i(\boldsymbol{x})|}\cdot\frac{\partial e_i(\boldsymbol{x})}{\partial\boldsymbol{x}}\right] \quad (42)$$

and the gradient of $e_i(\boldsymbol{x})$ is given by (28)–(36).

B. Evaluating Gradient by Transformation Jacobian

As an alternative, gradient can be evaluated using transformation Jacobian. Conventionally, the polynomials in $H(z)$ are expressed as

$$N_k(z) = z^2 + a_{1k}z + a_{0k} \quad (43)$$
$$D_k(z) = z^2 + d_{1k}z + d_{0k}. \quad (44)$$

If we define a parameter vector that collects H_0, τ and all a_{lk} and d_{lk} for $l = 0, 1$ and $k = 1,\cdots,K$ as

$$\boldsymbol{y} = [H_0\ \tau\ a_{01}\ a_{11}\cdots a_{0K}\ a_{1K}\ d_{01}\ d_{11}\cdots d_{0K}\ d_{1K}]^T \quad (45)$$

then the gradient of function e_i in (22) with respect to $\boldsymbol{x}$ can be expressed as

$$\nabla_x e_i(\boldsymbol{x}) = \boldsymbol{J}\nabla_y e_i(\boldsymbol{x}) \quad (46)$$

where $\nabla_y e_i(\boldsymbol{x})$ is the gradient of $e_i(\boldsymbol{x})$ with respect to $\boldsymbol{y}$, which is given by (28)–(31) and

$$\frac{\partial e_i}{\partial d_{0k}} = -\frac{F(\boldsymbol{x},e^{j\omega_i T})}{D_k(e^{j\omega_i T})} \quad (47)$$
$$\frac{\partial e_i}{\partial d_{1k}} = -\frac{e^{j\omega_i T}F(\boldsymbol{x},e^{j\omega_i T})}{D_k(e^{j\omega_i T})} \quad (48)$$

for $k = 1,\cdots,K$, and $\boldsymbol{J} = \partial\boldsymbol{y}/\partial\boldsymbol{x}$ is the *transformation Jacobian* that is a $(4K+2)\times(4K+2)$ matrix determined by the parameterization transformation used. In effect, $\boldsymbol{J}$ is a block–diagonal matrix given by

$$\boldsymbol{J} = \begin{bmatrix} \boldsymbol{I}_{2K+2} & & & \\ & \boldsymbol{J}_1 & & \boldsymbol{0} \\ & & \boldsymbol{J}_2 & \\ & \boldsymbol{0} & & \ddots \\ & & & \boldsymbol{J}_K \end{bmatrix} \quad (49)$$

$$\frac{\partial e_i(\boldsymbol{x})}{\partial H_0} = \frac{F(\boldsymbol{x},\omega_i)}{H_0} \quad (28)$$
$$\frac{\partial e_i(x)}{\partial\tau} = j\omega_i F_0(\omega_i) \quad (29)$$
$$\frac{\partial e_i(\boldsymbol{x})}{\partial a_{0k}} = \frac{F(\boldsymbol{x},\omega_i)}{N_k(e^{j\omega_i T})} \quad (30)$$
$$\frac{\partial e_i(\boldsymbol{x})}{\partial a_{1k}} = \frac{e^{j\omega_i T}F(\boldsymbol{x},\omega_i)}{N_k(e^{j\omega_i T})} \quad (31)$$

$$\frac{\partial e_i(\boldsymbol{x})}{\partial b_{0k}} = \begin{cases} \dfrac{(\delta-1)[1+\tanh(b_{1k})e^{j\omega_i T}]F(\boldsymbol{x},\omega_i)}{\cosh^2(b_{0k})p_2^{(\delta)}(e^{j\omega_i T},b_{0k},b_{1k})} & \text{for HTT} \quad (32a) \\ \dfrac{2(\delta-1)[\pi+2\tan^{-1}(b_{1k})e^{j\omega_i T}]F(\boldsymbol{x},\omega_i)}{\pi^2(1+b_{0k}^2)q_2^{(\delta)}(e^{j\omega_i T},b_{0k},b_{1k})}, & \text{for ATT} \quad (32b) \\ \dfrac{4(1-\delta)b_{0k}[(b_{1k}^2-1)e^{j\omega_i T}+(1-\delta)(1+b_{1k}^2)]F(\boldsymbol{x},\omega_i)}{(1+b_{1k}^2+b_{0k}^2)^2 r_2^{(\delta)}(e^{j\omega_i T},b_{0k},b_{1k})}, & \text{for MBT} \quad (32c) \end{cases}$$

$$\frac{\partial e_i(\boldsymbol{x})}{\partial b_{1k}} = \begin{cases} \dfrac{(\delta-1)[1+\tanh(b_{0k})e^{j\omega_i T}]F(\boldsymbol{x},\omega_i)}{\cosh^2(b_{1k})p_2^{(\delta)}(e^{j\omega_i T},b_{0k},b_{1k})}, & \text{for HTT} \quad (33a) \\ \dfrac{2(\delta-1)[\pi+2\tan^{-1}(b_{0k})e^{j\omega_i T}]F(\boldsymbol{x},\omega_i)}{\pi^2(1+b_{1k}^2)q_2^{(\delta)}(e^{j\omega_i T},b_{0k},b_{1k})}, & \text{for ATT} \quad (33b) \\ \dfrac{4(1-\delta)b_{1k}[(b_{0k}^2-1)e^{j\omega_i T}+(1-\delta)(1+b_{0k}^2)]F(\boldsymbol{x},\omega_i)}{(1+b_{1k}^2+b_{0k}^2)^2 r_2^{(\delta)}(e^{j\omega_i T},b_{0k},b_{1k})}, & \text{for MBT} \quad (33c) \end{cases}$$

$$\frac{\partial e_i}{\partial b_{0,K+1}} = \begin{cases} \dfrac{(\delta-1)F(\boldsymbol{x},\omega_i)}{\cosh^2(b_{0,K+1})p_1^{(\delta)}(e^{j\omega_i T},b_{0,K+1})}, & \text{for HTT} \quad (36a) \\ \dfrac{2(\delta-1)F(\boldsymbol{x},\omega_i)}{\pi(1+b_{0,K+1}^2)q_1^{(\delta)}(e^{j\omega_i T},b_{0,K+1})}, & \text{for ATT} \quad (36b) \\ \dfrac{4(\delta-1)b_{0,K+1}F(\boldsymbol{x},\omega_i)}{(b_{0,K+1}^2+1)^2 r_1^{(\delta)}(e^{j\omega_i T},b_{0,K+1})}, & \text{for MBT} \quad (36c) \end{cases}$$

where $\boldsymbol{I}_{2K+2}$ is the identity matrix of dimension $2K+2$, and $\boldsymbol{J}_k$ is the 2×2 matrix

$$\boldsymbol{J}_k = \begin{bmatrix} \dfrac{\partial d_{0k}}{\partial b_{0k}} & \dfrac{\partial d_{1k}}{\partial b_{0k}} \\ \dfrac{\partial d_{0k}}{\partial b_{1k}} & \dfrac{\partial d_{1k}}{\partial b_{1k}} \end{bmatrix}$$

which depends on the transformation used, shown in (50a)–(50c), at the bottom of this page.

As will be shown shortly, the transformation Jacobian $\boldsymbol{J}$ will be of use in the derivation of a closed-form formula for the Hessian matrix of the objective function $E(x)$ defined by (26) and $\Psi(\boldsymbol{x}, \lambda, \xi)$ defined by (37).

C. Hessian Matrix

The importance of having the Hessian matrix of the objective function calculated lies in the fact that if the Hessian matrix is positive definite, then the classic Newton method can be used to update the filter coefficient and the value of the objective function will be reduced at a second-order rate. If the Hessian matrix is not positive definite, then the modified Newton (MN) method can be used to find a good descent direction with a satisfactory reduction rate [10], [11]. The design efficiency of the MN method as compared to several quasi-Newton optimization algorithms was investigated in detail by Bose and Chen [23]. In what follows, we derive expressions of the Hessian matrix that can be used in the least pth and minimax optimization.

For the objective function $E(x)$ defined by (26), it follows from (27) that the Hessian matrix of $E(x)$ is given by

$$\nabla_x^2 E(\boldsymbol{x}) = p \sum_{i=1}^{M} w_i \boldsymbol{H}_i^{(e)}(\boldsymbol{x}) \tag{51}$$

with

$$\begin{aligned} \boldsymbol{H}_i^{(e)}(\boldsymbol{x}) &= (p-2)|e_i(\boldsymbol{x})|^{p-4}\mathrm{Re}[\overline{e}_i(\boldsymbol{x})\nabla_x e_i(\boldsymbol{x})]\mathrm{Re}[\overline{e}_x(\boldsymbol{x})\nabla_x^T e_i(\boldsymbol{x})] \\ &\quad + p|e_i(\boldsymbol{x})|^{p-2}\mathrm{Re}[\overline{e}_i(\boldsymbol{x})\nabla_x^2 e_i(\boldsymbol{x}) + \nabla_x \overline{e}_i(\boldsymbol{x})\nabla_x^T e_i(\boldsymbol{x})] \end{aligned} \tag{52}$$

where $\nabla_x e_i(\boldsymbol{x})$ is the gradient of $e_i(\boldsymbol{x})$ with respect to $\boldsymbol{x}$ given by (28)–(33) or (46)–(50), and $\nabla_x^2 e_i(\boldsymbol{x})$ is the Hessian matrix of $e_i(\boldsymbol{x})$ with respect to $\boldsymbol{x}$, which is to be derived shortly. For the least squares case, i.e., $p = 2$, (52) is quite simple:

$$\boldsymbol{H}_i^{(e)}(\boldsymbol{x}) = \mathrm{Re}[\overline{e}_i(\boldsymbol{x})\nabla_x^2 e_i(\boldsymbol{x}) + \nabla_x \overline{e}_i(\boldsymbol{x})\nabla_x^T e_i(\boldsymbol{x})]. \tag{53}$$

For the objective function $\Psi(\boldsymbol{x}, \lambda, \xi)$ defined by (37), it follows from (41) and (42) that

$$\begin{aligned} &\nabla_x^2 \psi(\boldsymbol{x}, \lambda_k, \xi) \\ &\quad = \xi \sum_{i \in I_1} \lambda_{ki} |e_i(\boldsymbol{x})|^{-2} \\ &\qquad \cdot \mathrm{Re}[\overline{e}_i(\boldsymbol{x})\nabla_x e_i(\boldsymbol{x})]\mathrm{Re}[\overline{e}_i(\boldsymbol{x})\nabla_x^T e_i(\boldsymbol{x})] \\ &\qquad + \sum_{i \in I_1} \lambda_{ki}[1 - \xi|e_i(\boldsymbol{x})|^{-1}]\boldsymbol{H}_i^{(e)}(\boldsymbol{x}) \\ &\qquad + \xi \sum_{i \in I_2} |e_i(\boldsymbol{x})|^{-2}\mathrm{Re}[\overline{e}_i(\boldsymbol{x})\nabla_x e_i(\boldsymbol{x})] \\ &\qquad \cdot \mathrm{Re}[\overline{e}_i(\boldsymbol{x})\nabla_x^T e_i(\boldsymbol{x})] + \sum_{i \in I_2} [1 - \xi|e_i(\boldsymbol{x})|^{-1}]\boldsymbol{H}_i^{(e)}(\boldsymbol{x}) \end{aligned} \tag{54}$$

where $\boldsymbol{H}_i^{(e)}(\boldsymbol{x})$ is given by (53). Note that the key quantity that needs to be formulated in order to evaluate $\nabla_x^2 E(\boldsymbol{x})$ and $\nabla_x^2 \psi(\boldsymbol{x}, \lambda_k, \xi)$ is $\nabla_x^2 e_i(\boldsymbol{x})$.

D. A Formula for $\nabla_x^2 e_i(\boldsymbol{x})$

From (48), we write

$$\begin{aligned} \nabla^2 e_i(\boldsymbol{x}) &= \nabla_x(\nabla_x^T e_i) = \nabla_x(\nabla_y^T e \boldsymbol{J}^T) \\ &= [\nabla_x(\nabla_y^T e_i \boldsymbol{\gamma}_1^T) \cdots \nabla_x(\nabla_y^T e_i \gamma_{4k+2}^T)] \end{aligned}$$

where $\boldsymbol{\gamma}_i$ denotes the ith row of the transformation Jacobian $\boldsymbol{J}$. Straightforward calculations then lead to

$$\nabla_x^2 e_i(x) = \boldsymbol{J}[\nabla_y^2 e_i(\boldsymbol{x})]\boldsymbol{J}^T + [\boldsymbol{D}_1 \nabla_y e \cdots \boldsymbol{D}_{4k+2}\nabla_y e] \tag{55}$$

where

$$\boldsymbol{D}_i = \frac{\partial \boldsymbol{J}}{\partial x_i} \tag{56}$$

$$\boldsymbol{J}_k = \begin{bmatrix} \dfrac{1-\delta}{\cosh^2(b_{0k})} & \dfrac{(1-\delta)\tanh(b_{1k})}{\cosh^2(b_{0k})} \\ 0 & \dfrac{(1-\delta)[1+\tanh(b_{0k})]}{\cosh^2(b_{1k})} \end{bmatrix}, \quad \text{for HTT} \tag{50a}$$

$$\boldsymbol{J}_k = \begin{bmatrix} \dfrac{2(1-\delta)}{\pi(1+b_{0k}^2)} & \dfrac{4(1-\delta)\tanh^{-1}(b_{1k})}{\pi^2(1+b_{0k}^2)} \\ 0 & \dfrac{2(1-\delta)[1+\tanh^{-1}(b_{0k})]}{\pi^2(1+b_{0k}^2)} \end{bmatrix}, \quad \text{for ATT} \tag{50b}$$

$$\boldsymbol{J}_k = \begin{bmatrix} \dfrac{-4(1-\delta)^2 b_{0k}(1+b_{1k}^2)}{(1+b_{1k}^2+b_{0k}^2)^2} & \dfrac{-4(1-\delta)b_{0k}(b_{1k}^2-1)}{(1+b_{1k}^2+b_{0k}^2)^2} \\ \dfrac{4(1-\delta)^2 b_{1k} b_{0k}^2}{(1+b_{1k}^2+b_{0k}^2)^2} & \dfrac{4(1-\delta)b_{1k}(b_{0k}^2+2)}{(1+b_{1k}^2+b_{0k}^2)^2} \end{bmatrix}, \quad \text{for MBT} \tag{50c}$$

and $\nabla_y^2 e_i(\boldsymbol{x})$ is the Hessian matrix of $e_i(\boldsymbol{x})$ with respect to $\boldsymbol{y}$. It follows from (49) that

$$\boldsymbol{D}_i = \boldsymbol{0}, \qquad \text{for } 1 \le i \le 2K+2 \tag{57a}$$

and

$$\boldsymbol{D}_i = \begin{bmatrix} \boldsymbol{0} & & & & & & \\ & \boldsymbol{0} & & \boldsymbol{0} & & & \\ & & \ddots & & & & \\ & & & \boldsymbol{0} & & & \\ & & & & \dfrac{\partial \boldsymbol{J}_l}{\partial x_i} & & \\ & & \boldsymbol{0} & & & \boldsymbol{0} & \\ & & & & & & \ddots \\ & & & & & & & \boldsymbol{0} \end{bmatrix},$$

$$\text{for } 2K+3 \le i \le 4K+2 \tag{57b}$$

where $\boldsymbol{J_l}$ is the 2×2 block in (49) with $l = (i-2K-2)/2$ for even i and $l = (i-2K-3)/2$ for odd i.

V. Design Examples

In this section, we present two design examples to illustrate the proposed method. The first example is to design a low-pass IIR filter to meet the following specifications:

1) normalized passband edge $\omega_p = 0.2$;
2) normalized stopband edge $\omega_a = 0.3$;
3) maximum passband ripple $\le$ 0.1 dB;
4) minimum stopband attenuation $\ge$ 45 dB;
5) maximum deviation in the group delay in the passband $\le$ 5%;
6) stability margin $\delta \ge 0.05$.

As will be seen below, one can apply the proposed methods to achieve these design specifications with a tenth-order IIR transfer function.

Although local solutions obtained by the proposed design algorithm with any initial points are always stable, the high degree of nonlinearity of (either the least pth type or the minimax type) objective function implies that many local solutions do not represent satisfactory designs. An effective way to find a good initial point is to apply the well-known balanced truncation method [24] to a higher order finite-impulse-response (FIR) filter that approximates the desired frequency response. Such an FIR transfer function can be obtained using a conventional design method [1]. The balanced truncation usually yields a reduced-order IIR transfer function with good approximation accuracy and guaranteed stability [25], [26]. For the purpose of our design, a 20th-order half-band linear-phase FIR filter was designed using the window method. A tenth-order IIR approximation of the FIR filter was obtained by the balanced truncation method. The three proposed parameterization methods were employed in the design, which was carried out using a quasi-Newton optimization method known as the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm [10], [11] and the MN method [10]. The set Ω contains 100 frequency values that are evenly placed in the normalized frequency band [0, 0.5]. The least pth objective function (69) was used with $p = 2$ and the weights

$$w_i = \begin{cases} 1, & \text{for } \le i \le 40, 59 \le i \le 100 \\ 0.75, & \text{for } i = 41, 58 \\ 0.5, & \text{for } i = 42, 57 \\ 0, & \text{for } 43 \le i \le 56. \end{cases}$$

TABLE I
Design Results of Example 1

Optimization algorithm	BFGS			Modified Newton		
Parameterization	HTT	ATT	MBT	HTT	ATT	MBT
Number of iterations	99	164	126	74	91	90
Flops (M)	66.69	99.46	81.09	147.93	166.61	172.07
Maximum modulus of poles	0.8715	0.8636	0.9139	0.9448	0.8496	0.9150
Maximum passband ripple (dB)	0.0818	0.0663	0.0311	0.0773	0.0389	0.0785
Minimum stopband attenuation (dB)	47.5156	49.3101	46.5222	48.4584	45.7446	46.9811
Maximum relative deviation of group delay in passband	4.19%	3.63%	4.90%	1.89%	4.78%	1.80%

To implement the BFGS algorithm, (28)–(33) were used to evaluate the gradient functions required. Evaluation of the Hessian matrices required by the MN algorithm was carried out using (51), (52), and (55)–(57). Convergence was claimed when the two-norm difference between two consecutive points obtained by the algorithm was less than a given tolerance ε. In all the designs, ε was set to be 10^{-6}. With the same initial point obtained by balanced truncation, both the BFGS and MN algorithms converge for all three parameterization methods. The design results are summarized in Table I.

It is observed that all six designs are stable and have met the design specifications. However, the design efficiency as well as the performance of filters obtained varies. In general, the BFGS algorithm requires more iterations than the MN algorithm does for the algorithm to converge, but the MN algorithm usually requires more floating-point operations per iteration as the Hessian matrix has to be evaluated. The six designs achieve similar minimum stopband attenuation, but the BFGS/MBT combination achieves the smallest maximum passband ripple (0.0311 dB) while the MN/HTT and MN/MBT combinations offer less than 2% maximum relative deviation of group delay in the passband. The amplitude response, passband ripple, and phase response of the tenth-order IIR filter obtained with BFGS/MBT are depicted in Figs. 4–6.

As the second example, we consider approximating the desired amplitude shown in Fig. 7 by a tenth-order stable IIR filter. Design problems of this kind may be encountered when one employs a singular-value-decomposition approach to designing a circular symmetric 2-D low-pass filter [17].

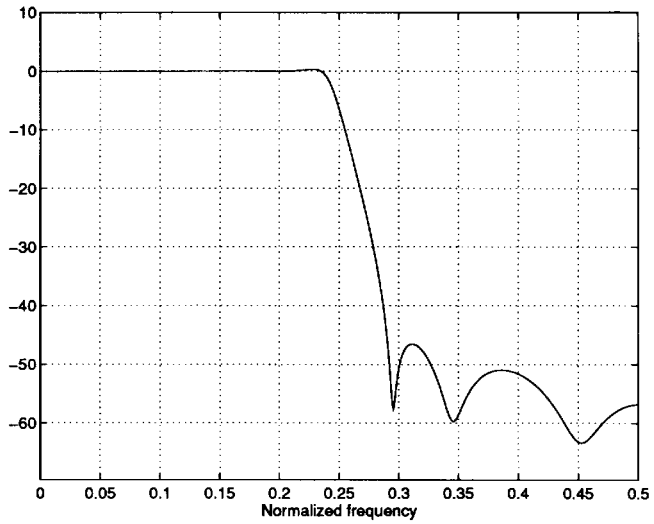


Fig. 4. Amplitude response of the tenth-order IIR filter (Example 1).

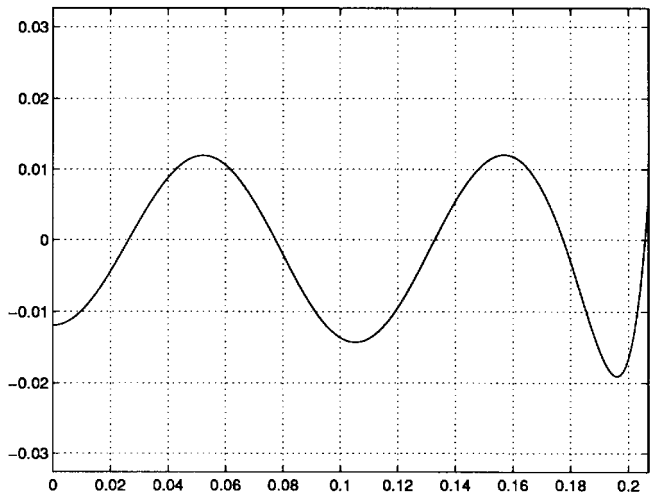
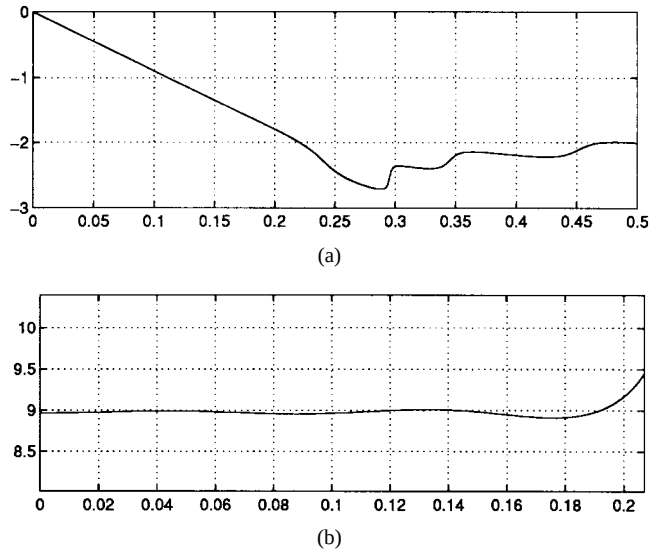


Fig. 5. Passband ripple of the filter (Example 1).



(a)

(b)

Fig. 6. Phase response of the filter (Example 1).

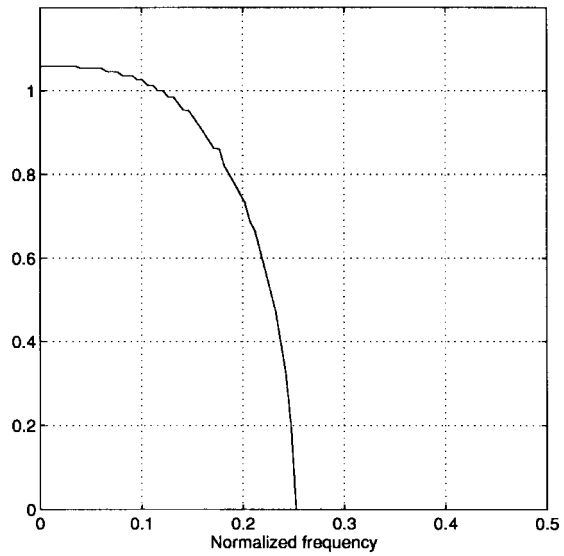


Fig. 7. Desired amplitude response (Example 2).

TABLE II
DESIGN RESULTS OF EXAMPLE 2

Optimization algorithm	BFGS			Modified Newton		
Parameterization	HTT	ATT	MBT	HTT	ATT	MBT
Number of iterations	205	156	216	93	93	93
Flops (M)	150.41	107.79	151.14	212.08	198.51	205.01
Maximum modulus of poles	0.9522	0.9348	0.9186	0.9592	0.9538	0.9200
E_2	0.0484	0.0487	0.0441	0.0437	0.0439	0.0440
Minimum stopband attenuation (dB)	29.7957	29.8884	30.0251	34.8266	34.5718	32.4356
Maximum relative deviation of group delay in passband	3.40%	3.04%	2.77%	4.04%	3.86%	2.97%

If the 2-D filter is required to have a linear phase response, then we need to design a stable IIR filter to approximate the desired magnitude response in Fig. 7, whose phase response in the passband is required to be linear. The BFGS and MN algorithms were used to carry out six designs where the proposed parameterization methods were employed to ensure stability of the designs. As the desired amplitude response is not a standard low-pass type, we use the two-norm measure

$$E_2 = \frac{1}{M}\sum_{i=1}^{M}[|F(\boldsymbol{x},\omega_i)| - |F_0(\omega_i)|]^2$$

to evaluate the approximation accuracy where $F(\boldsymbol{x},\omega_i)$ and $F_0(\omega_i)$ are defined by (23) and (24). With $p = 2, \varepsilon = 10^{-6}$ and the weights given by

$$w_i = \begin{cases} 2, & 49 \le i \le 54 \\ 1, & \text{elsewhere} \end{cases}$$

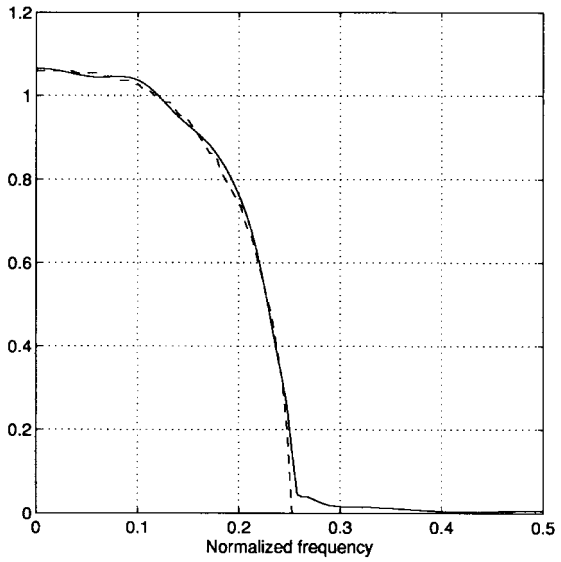


Fig. 8. Amplitude response of the IIR filter as compared to the desired one (Example 2).

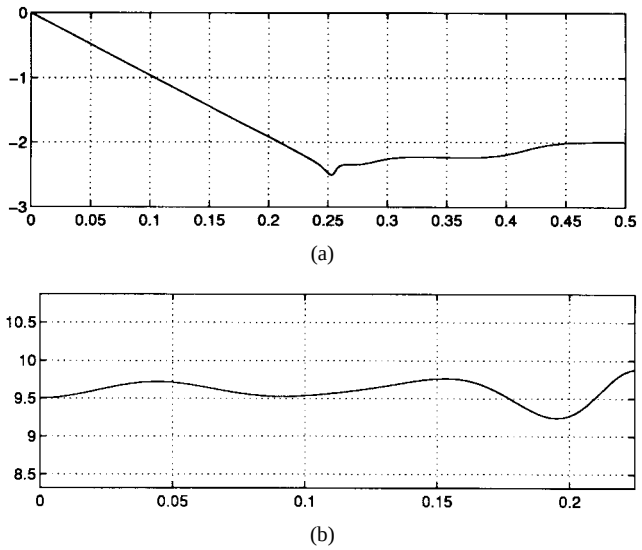


Fig. 9. Phase response of the filter (Example 2).

both the BFGS and MN algorithms converge, and the design results are summarized in Table II. It is observed that all the designs obtained are stable and have comparable performances with the MN/HTT combination achieving the smallest E_2 and largest stopband attenuation. As with the first example, the MN method requires less iterations, but more floating-point operations. Since digital filters are commonly designed off-line, computational complexity is not as important as the performance of the filter. In this regard, the MN method in conjunction with the proposed parameterization techniques deserve to be considered as "the method of choice." The amplitude and phase responses of the filter designed by MN/HTT approach are shown in Figs. 8 and 9.

VI. Concluding Remarks

We have proposed a parameterization approach to the design of IIR filters with a prescribed stability margin. The parameterization of all stable IIR transfer functions allows one to carry out the design by using well-established unconstrained optimization methods. The examples presented in Section V have demonstrated that stable low-order IIR filters with desirable amplitude/phase characteristics can effectively be designed using the proposed techniques. The concept of stable parameterization is expected to be of use in other digital signal-processing (DSP) problems, including adaptive IIR filtering [27], where filter coefficients are updated to achieve specific filtering tasks while preserving stability.

Acknowledgment

The author is grateful to the reviewers for their constructive comments, which have led to an improved version of this paper.

References

[1] A. Antoniou, *Digital Filters: Analysis and Design*, 2nd ed. New York: McGraw-Hill, 1993.
[2] K. Steiglitz, "Computer-aided design of recursive digital filters," *IEEE Trans. Audio Electroacoust.*, vol. AU-18, pp. 123–129, June 1970.
[3] A. G. Deczky, "Synthesis of recursive digital filters using the minimum p-error criterion," *IEEE Trans. Audio Electroacoust.*, vol. AU-20, pp. 257–263, Oct. 1972.
[4] J. W. Bandler and B. L. Bardakjian, "Least pth optimization of recursive digital filters," *IEEE Trans. Audio Electroacoust.*, vol. AU-21, pp. 460–470, Oct. 1973.
[5] A. G. Deczky, "Equiripple and minimax (Chebyshev) approximations for recursive digital filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-22, pp. 98–111, Apr. 1974.
[6] C. Charalambous, "Minimax optimization of recursive digital filters using recent minimax results," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-23, pp. 333–345, Aug. 1975.
[7] A. T. Chottera and G. A. Jullien, "A linear programming approach to recursive digital filter design with linear phase," *IEEE Trans. Circuits Syst.*, vol. 29, pp. 139–149, Mar. 1982.
[8] W.-S. Lu, S.-C. Pei, and C.-C. Tseng, "A weighted least squares method for the design of stable 1-D and 2-D IIR filters," *IEEE Trans. Signal Processing*, vol. 46, pp. 1–10, Jan. 1998.
[9] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*. New York: Academic, 1981.
[10] D. G. Luenberger, *Linear and Nonlinear Programming*, 2nd ed. Reading, MA: Addison-Wesley, 1984.
[11] R. Fletcher, *Practical Methods of Optimization*, 2nd ed. New York: Wiley, 1987.
[12] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*. Philadelphia, PA: SIAM, 1994.
[13] Y. Nesterov and A. Nemirovskii, *Interior-Point Polynomial Algorithms in Convex Programming*. Philadelphia, PA: SIAM, 1994.
[14] S.-C. Pei and J.-J. Shyu, "Design of 1-D and 2-D IIR eigenfilters," *IEEE Trans. Signal Processing*, vol. 42, pp. 962–966, Apr. 1994.
[15] S.-C. Pei and J.-J. Shyu, "Eigenfilter design of 1-D and 2-D IIR digital all-pass filters," *IEEE Trans. Signal Processing*, vol. 42, pp. 966–968, Apr. 1994.
[16] T. Q. Ngugen, J. I. Laakso, and R. D. Koilpillai, "Eigenfilter approach for the design of all-pass filters approximating a given phase response," *IEEE Trans. Signal Processing*, vol. 42, pp. 2257–2263, Sept. 1994.
[17] W.-S. Lu and A. Antoniou, "Design of 2-D digital filters with arbitrary amplitude and phase responses using singular-value decomposition," in *IEEE Proc. Symp. Circuits Syst.*, Singapore, June 1991, pp. 618–621.
[18] E. I. Jury, *Theory and Application of z-Transform Method*. New York: Wiley, 1964.
[19] M. J. D. Powell, "Restart procedures for the conjugate gradient method," *Math. Programming*, vol. 12, pp. 241–254, 1977.
[20] C. Charalambous, "Acceleration of the least pth algorithm for minimax optimization with engineering applications," *Math. Programming*, vol. 17, pp. 270–297, 1979.

[21] C. Charalambous and A. Antoniou, "Equalization of recursive digital filters," *Proc. Inst. Elect. Eng.*, vol. 127, pt. G, pp. 219–225, Oct. 1980.
[22] A. Antoniou, "Improved minimax optimization algorithms and their application in the design of recursive digital filters," *Proc. Inst. Elect. Eng.*, vol. 138, pt. G, pp. 724–730, Dec. 1991.
[23] T. Bose and M.-Q. Chen, "Design of two-dimensional digital filters in the spatial domain," *IEEE Trans. Signal Processing*, vol. 41, pp. 1464–1469, Mar. 1993.
[24] B. C. Moore, "Principal component analysis in linear systems: Controllability, observability and model reduction," *IEEE Trans. Automat. Contr.*, vol. AC-26, pp. 17–32, Feb. 1981.
[25] U. M. Al-Saggaf and G. F. Franklin, "Model reduction via balanced realization: An extension and frequency weighted techniques," *IEEE Trans. Automat. Contr.*, vol. 33, pp. 687–692, July 1988.
[26] L. Pernebo and L. M. Silverman, "Model reduction via balanced state space representations," *IEEE Trans. Automat. Contr.*, vol. AC-27, pp. 382–387, Apr. 1982.
[27] J. J. Shynk, "Adaptive IIR filtering," *IEEE Acoust., Speech, Signal Processing Mag.*, vol. 6, pp. 4–21, Apr. 1989.

Wu-Sheng Lu (S'81–M'85–SM'90) received the B.S. degree in mathematics from Fudan University, Fudan, China, in 1964, and the M.S. degree in electrical engineering and Ph.D. degree in control science from the University of Minnesota, Minneapolis, in 1983 and 1984, respectively.

He was a Post-Doctoral Fellow at the University of Victoria, Victoria, B.C., Canada, in 1985, and a Visiting Assistant Professor at the University of Minnesota, in 1986. Since 1987, he has been with the University of Victoria, where he is currently a Professor. His teaching and research interests are in the areas of DSP and robotics. He has co-authored *Two-Dimensional Digital Filters* (New York: Marcel Dekker, 1992). He was an Associate Editor of the *Canadian Journal of Electrical and Computer Engineering* (1989), as well as Editor (1990–1992). He is currently an Associate Editor of *Multidimensional Systems and Signal Processing*.

Dr. Lu was an associate editor of IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II (1993–1995).