Optimal Design of IIR Digital Filters With Robust Stability Using Conic-Quadratic-Programming Updates

Wu-Sheng Lu, *Fellow, IEEE,* and Takao Hinamoto, *Fellow, IEEE*

***Abstract*—In this paper, minimax design of infinite-impulse-response (IIR) filters with prescribed stability margin is formulated as a conic quadratic programming (CQP) problem. CQP is known as a class of well-structured convex programming problems for which efficient interior-point solvers are available. By considering factorized denominators, the proposed formulation incorporates a set of linear constraints that are sufficient and near necessary for the IIR filter to have a prescribed stability margin. A second-order cone condition on the magnitude of each update that ensures the validity of a key linear approximation used in the design is also included in the formulation and eliminates a line-search step. Collectively, these features lead to improved designs relative to several established methods. The paper then moves on to extend the proposed design methodology to quadrantally symmetric two-dimensional (2-D) digital filters. Simulation results for both one-dimensional (1-D) and 2-D cases are presented to illustrate the new design algorithms and demonstrate their performance in comparison with several existing methods.**

***Index Terms*—Conic quadratic programming, IIR digital filters, robust stability, 2-D IIR digital filters.**

I. Introduction

INFINITE-IMPULSE-RESPONSE (IIR) digital filters are useful in a wide range of applications where high selectivity and efficient processing of discrete signals are desirable [1]–[17].

A major problem encountered in the design of IIR filters is stability. In unconstrained-optimization-based methods, the stability can be taken into account by variable transformations that convert the finite stability region into the entire parameter space [13]. As a result, the designer must deal with an objective function of increased nonlinearity. A more recent trend is to treat the design problem in a constrained optimization setting, where the stability requirement is incorporated as linear positive realness of the denominator [9], [12], Rouché's condition on denominator perturbations [14], iterative Lyapunov inequality constraints [15], [16], or a general positive realness constraint on denominator perturbations [17]. A common drawback of the above approaches is that they are all sufficient but *not* necessary conditions for stability. Consequently, good design candidates may be excluded from the design process.

In this paper, we propose a new constrained optimization method for the minimax design of stable one-dimensional (1-D) and quadrantally symmetric two-dimensional (2-D) IIR digital filters (it is known that the transfer function of a quadrantally symmetric 2-D digital filter has separable denominators [29]). The design method has several features.

i) It unifies 1-D and 2-D IIR filter designs by performing a sequence of linear updates of the design variables with each update carried out in a conic quadratic programming (CQP) setting. CQP represents a class of well-structured convex programming problems for which efficient interior-point optimization solvers are available.

ii) In our design formulation, the transfer function has a factorized denominator for which the necessary *and* sufficient stability condition can be characterized as a set of linear inequality constraints on the denominator coefficients that in principle excludes no good design candidates and fits naturally into the CQP formulation.

iii) The above set of linear constraints can be readily modified to ensure a stability margin in terms of pole radius. The modified constraints remain linear, and they are sufficient and *near* necessary for the stability robustness. It should be mentioned that CQP-based methods for filter design were proposed in [19] and [20], but only FIR filters were considered while the focus of the present paper is on IIR filters, dealing with rational transfer functions and their robust stability.

The paper is organized as follows. Section II gives preliminaries on the stability triangle of second-order discrete-time systems and basic formulation of CQP. Section III presents an analysis on how an internal stability triangle (of a second-order system) is related to the pole radius of the system. In Section IV, we outline a CQP-based design formulation applicable to both 1-D and 2-D IIR filters. The design algorithms for 1-D and quadrantally symmetric 2-D filters with separable denominators are given in Sections V and VI, respectively, with design examples for performance evaluation in comparison to several established methods.

In the rest of the paper, boldfaced characters denote matrices and vectors, $\boldsymbol{I}_r$ denotes the identity matrix of dimension r, and $\|\cdot\|$ denotes the standard Euclidean norm; ω_p and ω_a denote normalized passband and stopband edges, respectively, and the

Manuscript received April 16, 2002; revised December 3, 2002. The associate editor coordinating the review of this paper and approving it for publication was Dr. Masaaki Ikehara.

W.-S. Lu is with the Department of Electrical and Computer Engineering, University of Victoria, Victoria, BC, V8W 3P6 Canada (e-mail: wslu@ece.uvic.ca).

T. Hinamoto is with the Graduate School of Engineering, Hiroshima University, Higashi-Hiroshima, 739-8527, Japan (e-mail: hinamoto@hiroshima-u.ac.jp).

Digital Object Identifier 10.1109/TSP.2003.811229

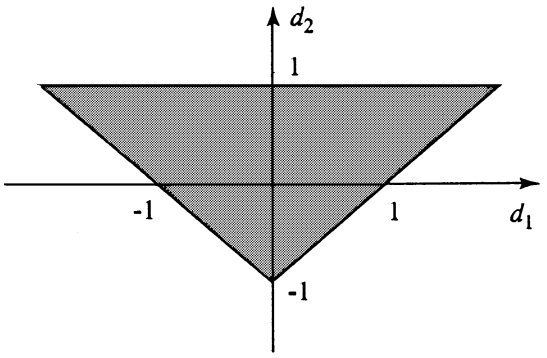


Fig. 1. Stability triangle.

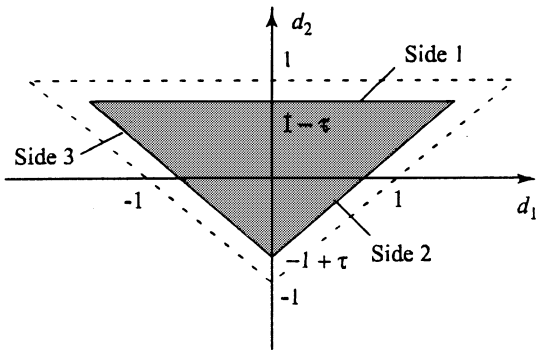


Fig. 2. Internal stability triangle.

normalized 1-D and 2-D base frequency bands are denoted by $\Omega = \{\omega\colon -\pi \le \omega \le \pi\}$ and $\Omega_2 = \{(\omega_1, \omega_2)\colon -\pi \le \omega_1, \omega_2 \le \pi\}$, respectively.

II. Preliminaries

A. Stability Triangle of Second-Order Systems

Let $H(z) = a(z)/d(z)$ be the transfer function of a second-order discrete-time system where

$$d(z) = z^2 + d_1 z + d_2. \tag{1}$$

It is well known that the system is stable if and only if coefficients d_1 and d_2 satisfy [1]

$$\boldsymbol{C}_2\boldsymbol{d} + \hat{\boldsymbol{e}} > \boldsymbol{0} \tag{2}$$

where

$$\boldsymbol{C}_2 = \begin{bmatrix} 1 & 1 \\ -1 & 1 \\ 0 & -1 \end{bmatrix}, \qquad \boldsymbol{d} = \begin{bmatrix} d_1 \\ d_2 \end{bmatrix}, \qquad \hat{\boldsymbol{e}} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}. \tag{3}$$

Note that the constraints in (2) are *linear* with respect to d_1 and d_2 and characterize the triangle in the (d_1, d_2)-space shown in Fig. 1, which will be referred to as the *stability triangle*.

B. Conic Quadratic Programming

Conic quadratic programming, which is sometimes called the second-order cone programming [18], [32], is a subclass of convex programming problems where a linear function is minimized subject to a set of second-order cone constraints [18], [20]:

$$\text{minimize} \quad \boldsymbol{f}^T\boldsymbol{x} \tag{4a}$$

$$\text{subject to:} \quad \|\boldsymbol{A}_i\boldsymbol{x} + \boldsymbol{b}_i\| \le \boldsymbol{c}_i^T\boldsymbol{x} + h_i, \qquad i = 1, 2 \ \ldots, N \tag{4b}$$

where $\boldsymbol{f} \in \Re^{n\times 1}$, $\boldsymbol{A}_i \in \Re^{(n_i-1)\times n}$, $\boldsymbol{b}_i \in \Re^{(n_i-1)\times 1}$, $\boldsymbol{c}_i \in \Re^{n\times 1}$, and $h_i \in \Re$. The term "conic" here reflects the fact that each constraint in (4b) is equivalent to a conic constraint

$$\begin{bmatrix} \boldsymbol{A}_i \\ \boldsymbol{c}_i^T \end{bmatrix}\boldsymbol{x} + \begin{bmatrix} \boldsymbol{b}_i \\ h_i \end{bmatrix} \in \mathcal{C}_i$$

where $\mathcal{C}_i$ is the second-order cone in $\Re^{n_i}$, i.e.,

$$\mathcal{C}_i = \left\{ \begin{bmatrix} \boldsymbol{u} \\ t \end{bmatrix} \colon \boldsymbol{u} \in \Re^{(n_i-1)\times 1},\ t \ge 0,\ \|\boldsymbol{u}\| \le t \right\}.$$

From (4), it is evident that CQP includes linear programming and convex quadratic programming as special cases. On the other hand, since each constraint in (4b) can be expressed as

$$\begin{bmatrix} (\boldsymbol{c}_i^T\boldsymbol{x} + h_i)\boldsymbol{I} & \boldsymbol{A}_i\boldsymbol{x} + \boldsymbol{b}_i \\ (\boldsymbol{A}_i\boldsymbol{x} + \boldsymbol{b}_i)^T & \boldsymbol{c}_i^T\boldsymbol{x} + h_i \end{bmatrix} \succeq \boldsymbol{0} \tag{5}$$

where $\boldsymbol{M} \succeq \boldsymbol{0}$ denotes that $\boldsymbol{M}$ is positive semidefinite, the CQP is a subclass of semidefinite programming (SDP) [20], [21]. Commercial and public domain software based on interior-point optimization algorithms for CQP and SDP are available [22]–[24]. It is important to stress, however, that in general, the problem in (4) can be solved more efficiently as a CQP problem than solving it in an equivalent SDP setting [18]. In the subsequent sections, we attempt to formulate the design problems at hand as CQP problems rather than SDP problems.

III. Relation of an Internal Stability Triangle to Pole Radius

Consider a second-order system whose transfer function is $H(z) = a(z)/d(z)$ with $d(z)$ given in (1). For the sake of robust stability, we consider a triangle in (d_1, d_2)-space that is strictly inside the stability triangle as shown in Fig. 2, where τ is a small positive scalar. The region enclosed with the internal triangle is characterized by

$$\boldsymbol{C}_2\boldsymbol{d} + (1-\tau)\hat{\boldsymbol{e}} \ge \boldsymbol{0} \tag{6}$$

where $\boldsymbol{C}_2$, $\boldsymbol{d}$, and $\hat{\boldsymbol{e}}$ are defined in (3). For a point (d_1, d_2) on side 1 of the internal triangle (see Fig. 2), we have $d_2 = 1-\tau$ and $-2+2\tau \le d_1 \le 2-2\tau$, and the associated poles are given by

$$p_{1,2} = \frac{-d_1 \pm j\sqrt{4(1-\tau) - d_1^2}}{2} \qquad \text{for } -2+2\tau \le d_1 \le 2-2\tau \tag{7}$$

whose magnitude is $\sqrt{1-\tau}$. When d_1 varies from $-2+2\tau$ to $2-2\tau$, it follows from (7) that side 1 corresponds to two separate partial circles of radius $\sqrt{1-\tau}$, which are shown as the solid

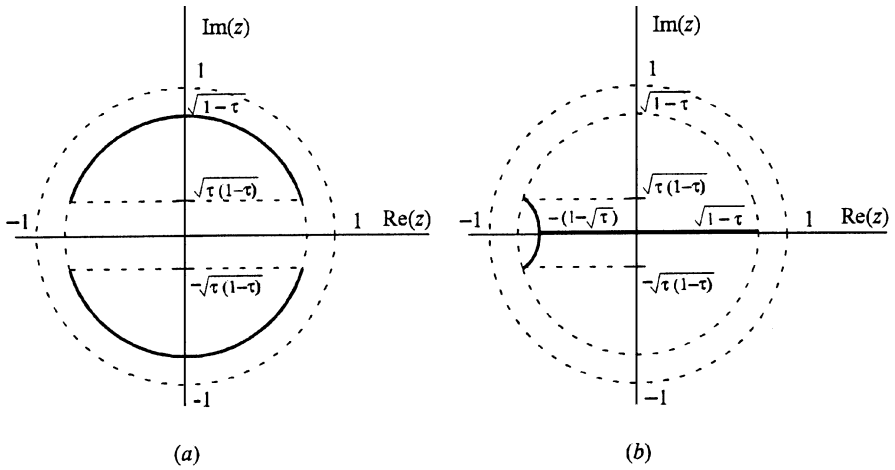


Fig. 3. Trajectories of the poles associated with (a) side 1 of the internal stability triangle and (b) side 2 of the triangle are shown as solid curves.

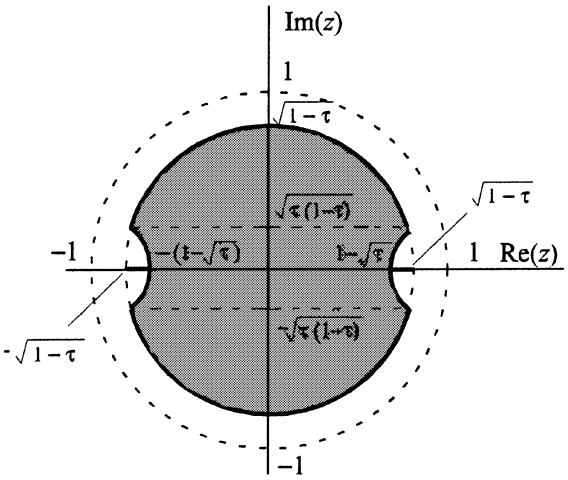


Fig. 4. Shaded region plus two short segments (in solid line) on the real-axis represent the pole locations corresponding to the internal stability triangle in Fig. 2.

curves in Fig. 3(a). For a point on side 2, we have $d_2 = d_1 - 1 + \tau$ for $0 \le d_1 \le 2 - 2\tau$, and the associated poles are given by

$$p_{1,2} = \begin{cases} \text{two reals with the largest being} \\ \dfrac{-d_1 + \sqrt{(d_1-2)^2 - 4\tau}}{2}, & \text{for } 0 \le d_1 \le 2 - 2\sqrt{\tau} \\ \dfrac{-d_1 \pm j\sqrt{4\tau - (d_1-2)^2}}{2}, & \text{for } 2 - 2\sqrt{\tau} < d_1 \le 2 - 2\tau. \end{cases} \tag{8}$$

Consequently, when d_1 varies from 0 to $2 - 2\tau$, the poles given by (8) generate the trajectory shown as the solid line in Fig. 3(b). Similarly, it can be readily verified that the poles associated with side 3 generate the mirror-image of the trajectory in Fig. 3(b) with respect to the vertical axis. Therefore, all system poles that are associated with the internal stability triangle in Fig. 2 are strictly inside the unit circle with a distance to the boundary no less than $1 - \sqrt{1-\tau}$. For a small $\tau > 0$ (which is always the case for filter design purposes), the poles cover a dominant part of the radius $-\sqrt{1-\tau}$ disk, as shown in Fig. 4. A polynomial is called a robust Schur polynomial with margin $\alpha > 0$ if the largest magnitude of its zeros is no larger than $1 - \alpha$. Thus, the second-order polynomial $d(z)$ in (1) is a robust Schur polynomial with margin $1 - \sqrt{1-\tau}$ if $\boldsymbol{d} = [d_1 \quad d_2]^T$ satisfies the constraint in (6).

IV. General Design Method Using Linear CQP Updates

In this section, we describe a general method that generates an optimal stable rational approximation for a given design specification in the minimax sense by using a sequence of linear updates for the design parameters with each update carried out in a CQP formulation. We will describe the method in a way that it is applicable to both 1-D and 2-D IIR filters. As such, our description will be given in a setting more general than each individual design algorithm whose algorithmic details will be presented in Sections V and VI.

Let $H(\omega, \boldsymbol{x})$ be a nonlinear function of frequency ω and parameter vector $\boldsymbol{x} \in \Re^{p\times 1}$, and let $H_d(\omega)$ be a desired function of ω on Ω. We seek to find a vector $\boldsymbol{x}$ that solves the constrained weighted minimax optimization problem

$$\underset{\boldsymbol{x}}{\text{minimize}}\{\underset{\omega\in\Omega}{\text{maximize}}\ W(\omega)|H(\omega, \boldsymbol{x}) - H_d(\omega)|\} \tag{9a}$$

$$\text{subject to:} \quad H(\omega, \boldsymbol{x}) \text{ stable} \tag{9b}$$

where the meaning of stability in (9b) will become apparent in Sections V and VI when (9) is related to a filter design problem.

If η denotes an upper bound of $W(\omega)|H(\omega, \boldsymbol{x}) - H_d(\omega)|$ on Ω, then the problem in (9) can be converted into

$$\text{minimize} \quad \eta \tag{10a}$$

$$\text{subject to:} \quad W(\omega)|H(\omega, \boldsymbol{x}) - H_d(\omega)| \le \eta \qquad \text{for } \omega \in \Omega \tag{10b}$$

$$H(\omega, \boldsymbol{x}) \text{ stable.} \tag{10c}$$

Suppose we have a reasonable initial point $\boldsymbol{x}_0$ to start, and we are now in the kth iteration. For a smooth $H(\omega, \boldsymbol{x})$ in a vicinity of point $\boldsymbol{x}_k$, we can write

$$H(\omega, \boldsymbol{x}_k + \boldsymbol{\delta}) \approx H(\omega, \boldsymbol{x}_k) + \boldsymbol{g}_k^T(\omega)\boldsymbol{\delta} \tag{11}$$

provided that

$$\|\boldsymbol{\delta}\| \text{ is small} \tag{12}$$

where $\boldsymbol{g}_k(\omega)$ is the gradient of $H(\omega, \boldsymbol{x})$ with respect to $\boldsymbol{x}$ and evaluated at $\boldsymbol{x}_k$. Thus, for $\boldsymbol{x} = \boldsymbol{x}_k + \boldsymbol{\delta}$ with $\boldsymbol{\delta}$ subject to (12), we have

$$\begin{aligned} W(\omega)&|H(\omega, \boldsymbol{x}) - H_d(\omega)| \\ &\approx W(\omega)|\boldsymbol{g}_k^T(\omega)\boldsymbol{\delta} + [H(\omega, \boldsymbol{x}_k) - H_d(\omega)]|. \end{aligned}$$

For filter design problems, $H(\omega, \boldsymbol{x}_k)$ and $H_d(\omega)$ are in general complex-valued, and we need to define

$$H(\omega, \boldsymbol{x}) = H_r(\omega, \boldsymbol{x}) + jH_i(\omega, \boldsymbol{x}) \tag{13a}$$

$$H_d(\omega) = H_{rd}(\omega) + jH_{id}(\omega) \tag{13b}$$

$$\boldsymbol{g}_k(\omega) = \boldsymbol{g}_{rk}(\omega) + j\boldsymbol{g}_{ik}(\omega). \tag{13c}$$

It follows that

$$\begin{aligned} W(\omega)&|H(\omega, \boldsymbol{x}) - H_d(\omega)| \\ &\approx |W(\omega)[\boldsymbol{g}_{rk}^T(\omega)\boldsymbol{\delta} + e_{rk}(\omega)] + jW(\omega)[\boldsymbol{g}_{ik}^T(\omega)\boldsymbol{\delta} + e_{ik}(\omega)]| \\ &= \|\boldsymbol{G}_k(\omega)\boldsymbol{\delta} + \boldsymbol{e}_k(\omega)\| \end{aligned} \tag{14}$$

where

$$\boldsymbol{G}_k(\omega) = W(\omega)\begin{bmatrix} \boldsymbol{g}_{rk}^T(\omega) \\ \boldsymbol{g}_{ik}^T(\omega) \end{bmatrix}, \qquad \boldsymbol{e}_k(\omega) = W(\omega)\begin{bmatrix} e_{rk}(\omega) \\ e_{ik}(\omega) \end{bmatrix}$$

$$e_{rk}(\omega) = H_r(\omega, \boldsymbol{x}_k) - H_{rd}(\omega)$$

$$e_{ik}(\omega) = H_i(\omega, \boldsymbol{x}_k) - H_{id}(\omega).$$

In the light of (10b), (12), and (14), we see that an approximate solution in the kth iteration can be obtained by solving the constrained optimization problem

$$\text{minimize} \quad \eta \tag{15a}$$

$$\text{subject to:} \quad \|\boldsymbol{G}_k(\omega)\boldsymbol{\delta} + \boldsymbol{e}_k(\omega)\| \le \eta \qquad \text{for } \omega \in \Omega \tag{15b}$$

$$\|\boldsymbol{\delta}\| \le \beta \tag{15c}$$

$$H(\omega, \boldsymbol{x}_k + \boldsymbol{\delta}) \text{ stable} \tag{15d}$$

where β is a prescribed bound to control the magnitude of $\boldsymbol{\delta}$. Once a solution of (15), say $\boldsymbol{\delta}_k$, is obtained, point $\boldsymbol{x}_k$ is updated to $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{\delta}_k$, and the kth iteration is claimed to be complete. The iteration process continues until $\|\boldsymbol{\delta}_k\|$ is less than a prescribed convergence tolerance ε. If we treat the upper bound η in (15a) and (15b) as an additional design variable and define an augmented parameter vector

$$\boldsymbol{u} = \begin{bmatrix} \eta \\ \boldsymbol{\delta} \end{bmatrix} \tag{16}$$

then the problem in (15) can be expressed as

$$\text{minimize} \quad \boldsymbol{c}^T\boldsymbol{u} \tag{17a}$$

$$\text{subject to:} \quad \|\hat{\boldsymbol{G}}_k(\omega)\boldsymbol{u} + \boldsymbol{e}_k(\omega)\| \le \boldsymbol{c}^T\boldsymbol{u} \qquad \text{for } \omega \in \Omega_d \tag{17b}$$

$$\|\hat{\boldsymbol{I}}\boldsymbol{u}\| \le \beta \tag{17c}$$

$$H(\omega, \boldsymbol{x}_k + \boldsymbol{\delta}) \text{ stable} \tag{17d}$$

where $\boldsymbol{c} = [1 \;\; 0 \;\; \cdots \;\; 0]^T$, $\hat{\boldsymbol{G}}_k(\omega)$ is generated by augmenting $\boldsymbol{G}_k(\omega)$ with a zero column on the left, $\hat{\boldsymbol{I}}$ is obtained by augmenting the identity matrix $\boldsymbol{I}_n$ with a zero column on the left, and $\Omega_d = \{\omega_i, 1 \le i \le K\} \subset \Omega$ is a set of dense grid points in the frequency region of interest.

As illustrated in Section III, a second-order section of an IIR filter possesses robust stability with its pole radius no larger than $\sqrt{1-\tau}$, as long as its denominator coefficients meet the linear constraint in (6). If $H(\omega, \boldsymbol{x}_k + \boldsymbol{\delta})$ represents the frequency response of an IIR digital filter whose denominator is factorized into a product of second-order sections (and a first-order section for odd-order denominators), then, as one may expect, the constraint in (17d) can be characterized by a set of linear inequality constrains as

$$\boldsymbol{C}\boldsymbol{u} + \boldsymbol{h} \ge \boldsymbol{0} \tag{18}$$

(see Section V for the structure of matrix $\boldsymbol{C}$ and vector $\boldsymbol{h}$).

Suppose matrix $\boldsymbol{C}$ has m rows; then, (17) can be expressed as

$$\boldsymbol{c}_i^T\boldsymbol{u} + h_i \ge 0, \qquad \text{for } 1 \le i \le m$$

where $\boldsymbol{c}_i$ is the ith column of $\boldsymbol{C}^T$, and h_i is the ith component of $\boldsymbol{h}$, and the problem in (17) becomes

$$\text{minimize} \quad \boldsymbol{c}^T\boldsymbol{u} \tag{19a}$$

$$\text{subject to} \quad \|\hat{\boldsymbol{G}}_k(\omega_i)\boldsymbol{u} + \boldsymbol{e}_k(\omega_i)\| \le \boldsymbol{c}^T\boldsymbol{u}, \qquad \text{for } 1 \le i \le K \tag{19b}$$

$$\|\hat{\boldsymbol{I}}\boldsymbol{u}\| \le \beta \tag{19c}$$

$$\boldsymbol{c}_i^T\boldsymbol{u} + h_i \ge 0, \qquad \text{for } 1 \le i \le m. \tag{19d}$$

On comparing the problem in (19) with that in (4), it is evident that problem (19) is a CQP problem with $p+1$ design variables, $K+1$ second-order cone constraints, and m linear constraints [obviously, a linear inequality constraint can be treated as a trivial second-order cone constraint; however, efficient CQP solvers (e.g., toolbox SeDuMi [22]) often deal with linear constraints and second-order cone constraints separately].

Several interior-point methods for CQP have been developed in the past; see, for example, [18] and [25]–[27]. Lucid exposition of the subject can be found in [20].

The original problem in (9) and, equivalently, the problem in (10) are highly nonlinear and nonconvex optimization problems. As such, the above method, if it converges, only provides a *local* minimizer for the problem. In general, the performance of such a local solution and the amount of computations required for the algorithm to converge depend largely on how an initial point is chosen. In the context of IIR filter design, however, it is found from our simulation study that the proposed design method is rather insensitive to the choice of the initial point. This issue will be addressed specifically in the next sections when design examples are presented. Concerning the convergence of the method, although a rigorous proof is presently not available, in our simulations, when the method was applied to design a variety of IIR

filters, we had not detected a single failure of convergence. One might attribute the success of the proposed method to two factors: i) The sub-problem involved in each iteration as formulated in (19) is a *convex* optimization problem for which globally convergent interior-point algorithms are available [18], [20], [32], and ii) we use constraint (19c), which validates the key approximation in (11).

Another related issue is the convergence rate or, in a more general term, the computational efficiency. From above description of the method, it is quite clear that the computational efficiency is determined by how efficient each individual SOCP problem in (19) is solved and how many linear updates are needed to reach a minimizer of (10). For the former, most of the algorithms that are presently available for solving SOCP problem (19) are so-called polynomial-time algorithms, meaning that the amount of computations required is bounded by a polynomial of the data size [20]. Consequently, the computational complexity for (19) is affordable for today¹s computing devices, even for designing high-order IIR filters, and it will increase only moderately when the size of the problem increases. For the latter, with a given bound b in constraint (19c), the number of updates needed depends on how far the initial point is from the minimizer. In the context of IIR filter design, the number of updates required is typically in the range of 15 to 50.

It should also be pointed out that although problem (19) is merely an *approximation* of (10), as the iteration continues and the local minimizer gets closer, the increment vector $\boldsymbol{\delta}$ obtained by solving (19) gradually shrinks in magnitude, and within a limited number of iterations, it eventually becomes such a value that the updated solution point is practically the same as the true minimizer.

V. Design of 1-D IIR Filters

A. Design Problem

Consider the transfer function of an IIR digital filter

$$H(z) = \frac{a(z)}{z^{n-r}d(z)} \tag{20a}$$

where

$$a(z) = \sum_{i=0}^{n} a_i z^{n-i}. \tag{20b}$$

$d(z)$ is a polynomial of order r expressed as product of second-order sections (and a first-order section if r is odd):

$$d(z) = \begin{cases} \displaystyle\prod_{i=1}^{r/2}(z^2 + d_{i1}z + d_{i2}), & \text{if } r \text{ even} \\ \displaystyle(z + d_o)\prod_{i=1}^{(r-1)/2}(z^2 + d_{i1}z + d_{i2}), & \text{if } r \text{ odd} \end{cases} \tag{20c}$$

and r is an integer between 0 and n. The reason our design formulation uses the above form of denominator, namely $z^{n-r}d(z)$, is that assigning a certain number of poles at the origin was found beneficial for the design of several types of digital filters, as observed in [14]. The design problem at hand is to determine the coefficients of $H(z)$ in (20) that solves the minimax optimization problem

$$\underset{\boldsymbol{x}}{\text{minimize}} \left[\underset{\omega\in\Omega}{\text{maximize}}\ W(\omega)|H(\omega, \boldsymbol{x}) - H_d(\omega)|\right] \tag{21a}$$

$$\text{subject to} \quad d(z) \neq 0, \qquad \text{for } |z| > \sqrt{1-\tau} \tag{21b}$$

where the filter coefficients form vector

$$\boldsymbol{x} = [a_0 \ \cdots \ a_n \ d_0 \ d_{11} \ d_{12} \ \cdots \ d_{L1} \ d_{L2}]^T$$

with L representing the number of second-order sections, i.e.,

$$L = \begin{cases} r/2, & \text{if } r \text{ even} \\ (r-1)/2, & \text{if } r \text{ odd} \end{cases} \tag{22}$$

(note that $\boldsymbol{x}$ contains component d_0 only if integer r is odd), $W(\omega) \geq 0$ is a weighting function on Ω, $H_d(\omega)$ is the desired frequency response, and $H(\omega, \boldsymbol{x})$ is the frequency response of the filter, which can be expressed as

$$H(\omega, \boldsymbol{x}) = \frac{a(\omega)}{d(\omega)} \tag{23}$$

with

$$a(\omega) = \boldsymbol{a}^T\boldsymbol{v}(\omega), \qquad \boldsymbol{a} = [a_0 \ a_1 \ \cdots \ a_n]^T$$

$$\boldsymbol{v}(\omega) = \boldsymbol{c}(\omega) - j\boldsymbol{s}(\omega)$$

$$\boldsymbol{c}(\omega) = [1 \ \cos\omega \ \cdots \ \cos n\omega]^T$$

$$\boldsymbol{s}(\omega) = [0 \ \sin\omega \ \cdots \ \sin n\omega]^T$$

$$d(\omega) = \begin{cases} \displaystyle\prod_{i=1}^{L}[1 + \boldsymbol{d}_i^T\boldsymbol{v}_2(\omega)], & \text{if } r \text{ even} \\ \displaystyle[1 + d_0 v_1(\omega)]\prod_{i=1}^{L}[1 + \boldsymbol{d}_i^T\boldsymbol{v}_2(\omega)], & \text{if } r \text{ odd} \end{cases}$$

$$v_1(\omega) = \cos\omega - j\sin\omega$$

$$\boldsymbol{d}_i = \begin{bmatrix} d_{i1} \\ d_{i2} \end{bmatrix}, \qquad \boldsymbol{v}_2(\omega) = \begin{bmatrix} \cos\omega \\ \cos 2\omega \end{bmatrix} - j\begin{bmatrix} \sin\omega \\ \sin 2\omega \end{bmatrix}.$$

The constraint in (21b) characterizes the requirement of robust stability that the pole radius of the filter be $\sqrt{1-\tau}$. On comparing (21) with (9), it is quite clear that the design can be accomplished using a sequence of linear updates, i.e., $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{\delta}_k$ for $k = 0, 1, \ldots$ with $\boldsymbol{\delta}_k$ solving the CQP problem in (19).

In order to implement the constraints in (19b), the gradient of $H(\omega, \boldsymbol{x})$ needs to be evaluated, and to implement the constraints in (19d), the robust stability constraint in (21b) has to be explicitly specified in terms of the design parameters. These two issues are addressed next.

B. Gradient of $H(\omega, \boldsymbol{x})$

Parameter vector $\boldsymbol{x}$ can be expressed in terms of vectors $\boldsymbol{a}$ and $\boldsymbol{d}_i$ defined in (23) as

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{a} \\ \boldsymbol{d} \end{bmatrix} = \begin{bmatrix} \boldsymbol{a} \\ d_0 \\ \boldsymbol{d}_1 \\ \vdots \\ \boldsymbol{d}_L \end{bmatrix} \begin{matrix} \} \, n+1 \text{ components} \\ \Big\} \, r \text{ components} \end{matrix} \tag{24}$$

where $\boldsymbol{d} = [d_0 \;\; \boldsymbol{d}_1^T \;\; \cdots \;\; \boldsymbol{d}_L^T]^T$ with component d_0 present only if r is odd. Using (23), the gradient of $H(\omega, \boldsymbol{x})$ with respect to $\boldsymbol{x}$ is evaluated as

$$\boldsymbol{g}(\omega, \boldsymbol{x}) = \begin{bmatrix} \dfrac{\partial H(\omega, \boldsymbol{x})}{\partial \boldsymbol{a}} \\ \dfrac{\partial H(\omega, \boldsymbol{x})}{\partial d_0} \\ \dfrac{\partial H(\omega, \boldsymbol{x})}{\partial \boldsymbol{d}_1} \\ \vdots \\ \dfrac{\partial H(\omega, \boldsymbol{x})}{\partial \boldsymbol{d}_L} \end{bmatrix} \tag{25}$$

with

$$\frac{\partial H(\omega, \boldsymbol{x})}{\partial \boldsymbol{a}} = \frac{\boldsymbol{v}(\omega)}{d(\omega)} \tag{26a}$$

$$\frac{\partial H(\omega, \boldsymbol{x})}{\partial d_0} = -H(\omega, \boldsymbol{x}) \frac{v_1(\omega)}{1 + d_0 v_1(\omega)} \tag{26b}$$

$$\frac{\partial H(\omega, \boldsymbol{x})}{\partial \boldsymbol{d}_i} = -H(\omega, \boldsymbol{x}) \frac{\boldsymbol{v}_2(\omega)}{1 + \boldsymbol{d}_i^T \boldsymbol{v}_2(\omega)} \qquad \text{for } 1 \le i \le L \tag{26c}$$

where $v_1(\omega)$, $\boldsymbol{v}_2(\omega)$, and $\boldsymbol{v}(\omega)$ are defined in (23). Note that $v_1(\omega)$ and $\boldsymbol{v}_2(\omega)$ are just the second and the second-and-third components of vector $\boldsymbol{v}(\omega)$, and they are all independent of the filter coefficients. It follows from (26) that if the frequency response $H(\omega, \boldsymbol{x})$ has been evaluated with its denominator sections stored separately, then the gradient can be evaluated immediately, as long as $\boldsymbol{v}(\omega)$ is available. Since the same $\boldsymbol{v}(\omega)$ can be used in every iteration, it is worthwhile to compute and store $\boldsymbol{v}(\omega)$ for $\omega \in \Omega_d$ [see (17b)] as a part of data preparation.

C. Constraints for Robust Stability

Suppose that point $\boldsymbol{x}_k$ represents a stable design and that the next point $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{\delta}_k$ is required to remain stable. Let

$$\boldsymbol{x}_k + \boldsymbol{\delta} = \begin{bmatrix} \boldsymbol{a} + \boldsymbol{\delta}_a \\ \boldsymbol{d} + \boldsymbol{\delta}_d \end{bmatrix} \tag{27}$$

and note that only vector $\boldsymbol{d} + \boldsymbol{\delta}_d$ effects the stability of the filter in question. For description convenience, we assume r is an odd integer so that vector $\boldsymbol{d} + \boldsymbol{\delta}_d$ assumes the form

$$\boldsymbol{d} + \boldsymbol{\delta}_d = \begin{bmatrix} d_0 + \delta_0 \\ \boldsymbol{d}_1 + \boldsymbol{\delta}_1 \\ \vdots \\ \boldsymbol{d}_L + \boldsymbol{\delta}_L \end{bmatrix} \tag{28}$$

where the first component is associated with the only first-order section in $d(z)$ whose robust stability is ensured if

$$-1 + \tau \le d_0 + \delta_0 \le 1 - \tau$$

i.e.,

$$\begin{bmatrix} 1 \\ -1 \end{bmatrix} (d_0 + \delta_0) + (1 - \tau) \begin{bmatrix} 1 \\ 1 \end{bmatrix} \ge \boldsymbol{0}. \tag{29a}$$

Each vector $\boldsymbol{d}_i + \boldsymbol{\delta}_i$ is connected to a second-order section in $d(z)$ whose robust stability is satisfied if (6) is imposed upon, i.e.,

$$\boldsymbol{C}_2(\boldsymbol{d}_i + \boldsymbol{\delta}_i) + (1 - \tau)\hat{\boldsymbol{e}} \ge \boldsymbol{0}, \qquad \text{for } 1 \le i \le L \tag{29b}$$

where $\boldsymbol{C}_2$ and $\hat{\boldsymbol{e}}$ are defined in (3). Therefore, $\boldsymbol{x}_k + \boldsymbol{\delta}$ in (27) represents an IIR filter with stability margin $1 - \sqrt{1 - \tau}$ if

$$\hat{\boldsymbol{C}}(\boldsymbol{d} + \boldsymbol{\delta}_d) + (1 - \tau)\boldsymbol{e} \ge \boldsymbol{0} \tag{30}$$

where $\boldsymbol{e} = [1 \;\; \cdots \;\; 1]^T \in R^{m \times 1}$ with $m = 3L + 2$, and

$$\hat{C} = \begin{bmatrix} \boldsymbol{c}_1 & & \boldsymbol{0} \\ & \boldsymbol{C}_2 & \\ & & \ddots \\ \boldsymbol{0} & & \boldsymbol{C}_2 \end{bmatrix}_{m \times r}$$

with $\boldsymbol{c}_1 = [1 \;\; -1]^T$ (if r is even, then the top-left $\boldsymbol{c}_1$ in $\hat{\boldsymbol{C}}$ is not present, and $m = 3L$). Now, if we augment matrix $\hat{\boldsymbol{C}}$ in (30) with $n+1$ columns of zeros on the left and replace $\boldsymbol{d} + \boldsymbol{\delta}_d$ with $\boldsymbol{x}_k + \boldsymbol{\delta}$, then (30) becomes

$$[\boldsymbol{0} \;\; \hat{\boldsymbol{C}}](\boldsymbol{x}_k + \boldsymbol{\delta}) + (1 - \tau)\boldsymbol{e} \ge \boldsymbol{0}$$

i.e.,

$$[\boldsymbol{0} \;\; \hat{C}]\boldsymbol{\delta} + \boldsymbol{h} \ge \boldsymbol{0} \tag{31a}$$

where

$$\boldsymbol{h} = [\underbrace{\boldsymbol{0}}_{(n+1) \text{ columns}} \;\; \hat{\boldsymbol{C}}]\boldsymbol{x}_k + (1 - \tau)\boldsymbol{e}. \tag{31b}$$

Finally, by augmenting the matrix in (31a) with one more zero column on the left and replacing vector $\boldsymbol{\delta}$ with $\boldsymbol{u}$ [defined in (16)], the stability constraint in (31) becomes

$$\boldsymbol{C}\boldsymbol{u} + \boldsymbol{h} \ge \boldsymbol{0} \tag{32}$$

where

$$\boldsymbol{C} = [\underbrace{\boldsymbol{0}}_{n+2 \text{ columns}} \;\; \hat{\boldsymbol{C}}].$$

Equivalently, (32) can be expressed as m linear inequality constraints, as seen in (19d), where $\boldsymbol{c}_i$ denotes the ith column of matrix $\boldsymbol{C}^T$ and h_i is the ith component of $\boldsymbol{h}$.

D. Design Algorithm

Given filter order (n, r), desired frequency response $H_d(\omega)$, and parameter τ, which is related to stability margin as $1 - \sqrt{1-\tau}$, the steps that carry out the proposed design method can be summarized as follows.

Step 1) *Data preparation:* This includes i) choosing an initial stable design $\boldsymbol{x}_0$ and a set of grid points $\Omega_d = \{\omega_i, 1 \le i \le K\}$; ii) evaluating vector $\boldsymbol{v}(\omega)$ [see (23)] on Ω_d; and iii) set iteration counter $k = 0$, bound β, and convergence tolerance ε. A straightforward choice of $\boldsymbol{x}_0$ assumes the form

$$\boldsymbol{x}_0 = \begin{bmatrix} \boldsymbol{a}_0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{33}$$

where $\boldsymbol{a}_0$ is the impulse response of an FIR filter of length $n+1$ that approximates $H_d(\omega)$. From (20), it is clear that the choice in (33) means that $d(z) = z^n$; hence, $H(z)$ is an FIR filter which is of course stable. Concerning item ii), typically, the number of grid points in Ω_d is in the range of 400 to 800, and the dimension of $\boldsymbol{v}(\omega)$ is usually in the range of 10 to 50. Thus, the data size shall not exceed 50 K, which is fairly moderate for today's PCs to store.

Step 2) At $\boldsymbol{x}_k$, solve the CQP problem in (19) for

$$\boldsymbol{u}_k = \begin{bmatrix} \eta_k \\ \boldsymbol{\delta}_k \end{bmatrix}$$

where $\boldsymbol{G}_k$ and $\boldsymbol{e}_k$ are defined in (14) with $\boldsymbol{g}_k$ evaluated using (26), the constraints in (19d) are specified by (32), and for typical bandpass filters, $W(\omega)$ is a piecewise-constant function of the form

$$W(\omega) = \begin{cases} 1, & \text{for } \omega \text{ in passband} \\ w, & \text{for } \omega \text{ in stopband} \\ 0, & \text{elsewhere.} \end{cases}$$

Step 3) Update the design from $\boldsymbol{x}_k$ to $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{\delta}_k$. If $\|\boldsymbol{\delta}_k\| < \varepsilon$, stop; otherwise, set $k := k+1$, and repeat from Step 2.

Several remarks are now in order. First, as the core of the algorithm, a CQP solver is called for computing each updating vector $\boldsymbol{\delta}_k$. Since an interior-point CQP solver is in general more efficient than its SDP counterpart, this leads to improved computational complexity compared with several SDP-based design methods [15], [16]. Second, incorporating the constraint in (19c) into the design formulation ensures the validity of the linear approximation (11) and eliminates the need for a line search step that is typically required in a nonlinear optimization algorithm. In a certain sense, constraint (19c) can be viewed as a trust-region strategy [28] that fits nicely into the current CQP formulation. Third, although the constraint in (2) is both sufficient and necessary for a second-order system to be stable, the robust stability constraint in (6) is only sufficient but *not* necessary for a second-order system to have a $\sqrt{1-\tau}$ pole radius. From the analysis in Section III (see Fig. 4), however, for a small τ, the stability constraints in the current design formulation is *near* necessary. Consequently, good design candidates are less likely to be excluded by the algorithm compared with the existing methods [9], [12], [14], [15], [17]. Finally, it should be mentioned that the idea of using a linear approximation for IIR filter design was initiated in [14], although the present design framework differs from that of [14] in terms of the use of a CQP formulation, the treatment of stability constraints, and the inclusion of a conic constraint on $\|\boldsymbol{\delta}\|$.

E. Design Example

A well-known IIR design is the minimax IIR lowpass filter of order $(n, r) = (12, 12)$ presented as Example 1 in Deczky [5], which has been used by many authors as a "benchmark filter" for comparison purposes. With $\omega_p = 0.5\pi$, $\omega_a = 0.6\pi$, and passband group delay $D = 15.9$ samples, the performance of the Deczky filter is shown in Fig. 5 (dash-dotted curves) and Table I. The proposed method was applied to design an IIR filter of order $(n, r) = (12, 12)$ with the same design parameters as specified above. The toolbox SeDuMi 1.05 [22] was used to implement the design algorithm on a 866 MHz Pentium III PC.

Two distinct initial points were tried. The first initial point $\boldsymbol{x}_0^{(1)}$ was obtained by designing an linear-phase FIR filter of length 33 using MATLAB function `fir1` and then applying balanced order reduction method [33] to obtain a stable IIR filter of order (12, 12). The second initial point $\boldsymbol{x}_0^{(2)}$ corresponds to a trivial IIR transfer function of the form $a(z)/z^{12}$, where $a(z)$ was obtained by simply designing linear-phase FIR filter of length 13 using MATLAB function `fir1`. Obviously, $\boldsymbol{x}_0^{(1)}$ was a considerably better initial point because its frequency response is much "closer" to the desired frequency response. With $\varepsilon = 5 \times 10^{-10}$, $K = 600$, $\tau = 0.05$, $b = 0.005$, $w = 1$, and initial point $\boldsymbol{x}_0^{(1)}$, the algorithm converged in 16 iterations with 473.23 Mflops and 66.42 s of CPU time. The performance of the IIR filter designed are evaluated in terms of the following.

- Error of frequency response in passband:

$$e_{fp}(\omega) = |H(e^{j\omega}) - H_d(e^{j\omega})| \qquad \text{for } \omega \in [0, \omega_p].$$

- Passband magnitude ripple:

$$e_{mp}(\omega) = |H(e^{j\omega})| - 1 \qquad \text{for } \omega \in [0, \omega_p].$$

- Stopband attenuation:

$$A(\omega) = 20 \log_{10} |H(e^{j\omega})| \qquad \text{for } \omega \in [\omega_a, \pi].$$

- Deviation in passband group delay

$$G(\omega) = \frac{\arg[H(e^{j\omega})] - D}{D} \qquad \text{for } \omega \in [0, \omega_p].$$

- Maximum magnitude of the poles.

The amplitude responses of the IIR filter obtained and the Deczky filter, their $e_{fp}(\omega)$, $e_{mp}(\omega)$, $A(\omega)$, and passband group

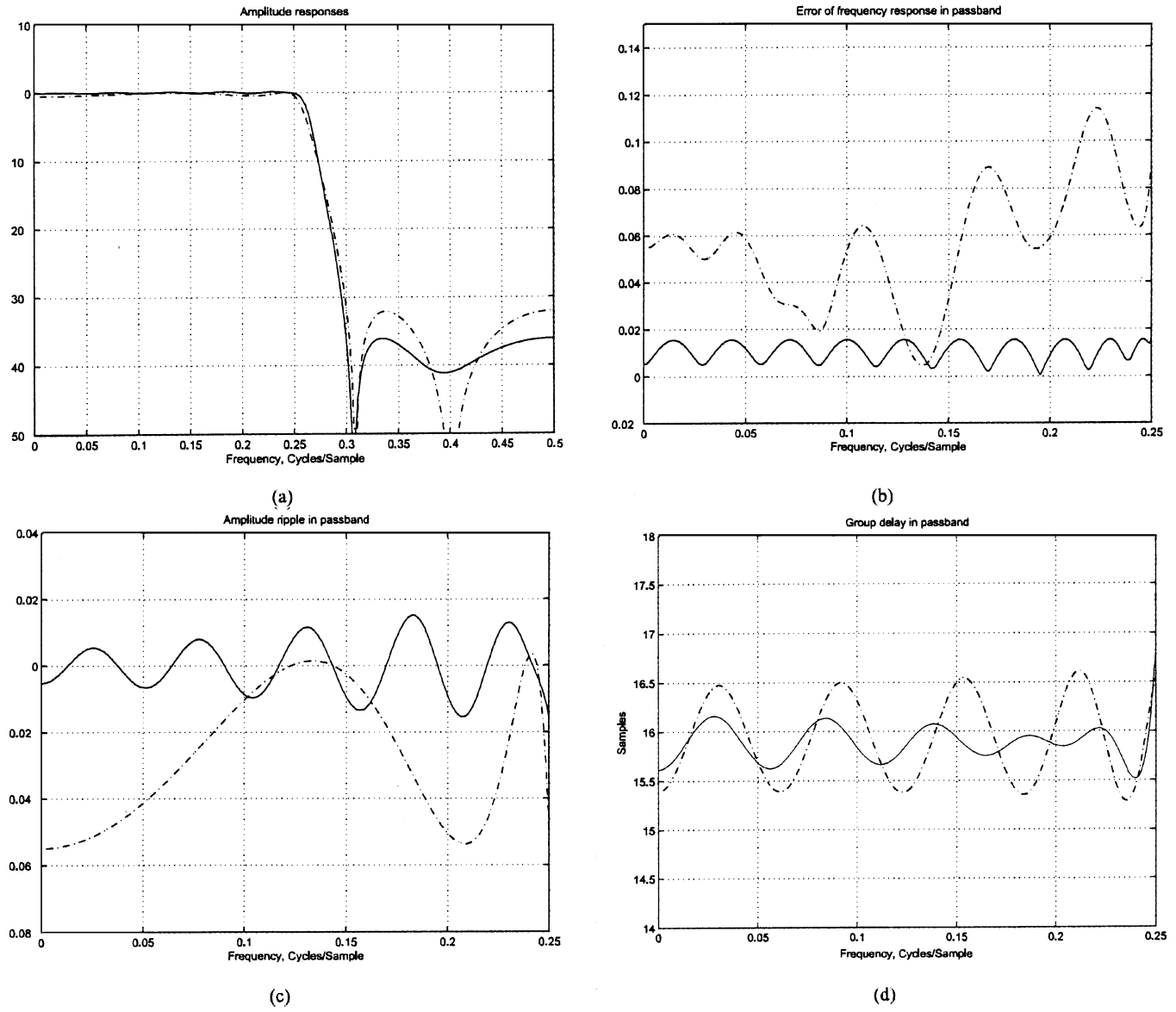


Fig. 5. (a) Amplitude responses. (b) Error of frequency response in passband. (c) Passband amplitude responses. (d) Passband group delays of the proposed design (solid curves) and the Deczky filter (dash-dotted curves).

delay are shown in Fig. 5, and $\max[e_{fp}(\omega)]$, $\max[|e_{mp}(\omega)|]$, $\min[A(\omega)]$, and average $G(\omega)$, i.e.,

$$\frac{1}{\omega_p}\int_0^{\omega_p} |G(\omega)|\, d\omega$$

are given in Table I. From Fig. 5 and Table I, considerable performance improvement over the Deczky filter were observed. The coefficients of the IIR filter designed are given in Table II.

It is worthwhile to report that with initial point $\boldsymbol{x}_0^{(2)}$, the proposed algorithm converged to the same solution point after 47 iterations. More iterations were expected because $\boldsymbol{x}_0^{(2)}$ is far away from the solution in comparison with $\boldsymbol{x}_0^{(1)}$.

VI. Design of 2-D IIR Filters

A. Quadrantally Symmetric 2-D IIR Filters

The class of quadrantally symmetric (QS) 2-D filters, together with their rotated counterparts, includes circularly symmetric filters, fan filters, and diamond-shaped filters, and covers practically all types linear 2-D filters that have been found useful in multidimensional signal processing. An important property of this class of 2-D filter is that a QS IIR transfer function always has *separable* denominators [29]. For the sake of notation simplicity, in this section, we are primarily concerned with circularly symmetric and diamond-shaped filters whose transfer functions assume the form

$$H(z_1, z_2) = \frac{a(z_1, z_2)}{(z_1 z_2)^{n-r} d(z_1) d(z_2)} \tag{34}$$

where

$$a(z_1, z_2) = \sum_{i=0}^{n}\sum_{k=0}^{n} a_{ik} z_1^{n-i} z_2^{n-k}$$

and $d(z)$ is defined by (20c). However, with straightforward modifications, the design methodology outlined below also

TABLE I
PERFORMANCE COMPARISON

IIR Filter of Order (n, r)	Deczky (12, 12)	Proposed (12, 12)
maximum error of frequency response in passband	0.1141	0.0156
maximum passband magnitude ripple	0.0549	0.0156
minimum stopband attenuation (dB)	31.7603	36.1455
passband group delay (sample)	15.9	15.9
average deviation in passband group delay	0.0233	0.0087
maximum magnitude of poles	0.8929	0.9220

TABLE II
COEFFICIENTS OF THE IIR FILTER

Numerator Coefficients	Denominator Coefficients
8.2583627e−3	1.00000e+0
−3.7603038e−2	−4.6514425e+0
8.6770226e−2	1.1950929e+1
−1.3388039e−1	−2.1706294e+1
1.6456835e−1	3.0426434e+1
−2.0044464e−1	−3.4240879e+1
2.7317099e−1	3.1473101e+1
−3.6702655e−1	−2.3702998e+1
4.2135950e−1	1.4494060e+1
−4.0569087e−1	−7.0264398e+0
3.5057805e−1	2.5743564e+0
−2.7393576e−1	−6.4541952e−1
1.4410970e−1	8.4986970e−2

applies to other types of QS filters. For notation simplicity, throughout the section, we denote the order of the 2-D IIR filter in (34) by (n, r). From (34), the frequency response of the filter can be expressed as

$$H(\omega_1, \omega_2) = \frac{a(\omega_1, \omega_2)}{d(\omega_1)d(\omega_2)} \tag{35a}$$

where

$$a(\omega_1, \omega_2) = \sum_{i=0}^{n}\sum_{k=0}^{n} a_{ik}e^{-j(i\omega_1+k\omega_2)} \tag{35b}$$

and $d(\omega)$ is given in (23). Because of the quadrantal symmetry of the filter, matrix $\{a_{ik},\ i, k = 0, 1, \ldots, n\}$ is symmetric, which means that $a(\omega_1, \omega_2)$ contains only $(n+1)(n+2)/2$ design parameters and can be expressed as

$$a(\omega_1, \omega_2) = \sum_{k=0}^{n} a_{kk}e^{-jk(\omega_1+\omega_2)} + \sum_{i=1}^{n}\sum_{k=0}^{i-1} a_{ik}[e^{-j(i\omega_1+k\omega_2)} + e^{-j(k\omega_1+i\omega_2)}]$$

which gives

$$a(\omega_1, \omega_2) = \boldsymbol{a}^T\boldsymbol{e}(\omega_1, \omega_2) \tag{36}$$

with

$$\boldsymbol{a} = \begin{bmatrix} a_{00} \\ \vdots \\ a_{nn} \\ a_{10} \\ a_{20} \\ a_{21} \\ \vdots \\ a_{n,n-1} \end{bmatrix}_{\hat{n}\times 1}$$

$$\boldsymbol{e}(\omega_1, \omega_2) = \begin{bmatrix} 1 \\ \vdots \\ e^{-jn(\omega_1+\omega_2)} \\ e^{-j\omega_1} + e^{-j\omega_2} \\ e^{-j2\omega_1} + e^{-j2\omega_2} \\ e^{-j(2\omega_1+\omega_2)} + e^{-j(\omega_1+2\omega_2)} \\ \vdots \\ e^{-j[n\omega_1+(n-1)\omega_2]} + e^{-j[(n-1)\omega_1+n\omega_2]} \end{bmatrix}$$

and $\hat{n} = (n+1)(n+2)/2$. This leads (35a) to

$$H(\omega_1, \omega_2, \boldsymbol{x}) = \frac{\boldsymbol{a}^T\boldsymbol{e}(\omega_1, \omega_2)}{d(\omega_1)\,d(\omega_2)} \tag{37}$$

where vector $\boldsymbol{x}$ collects the filter coefficients:

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{a} \\ d_0 \\ \boldsymbol{d}_1 \\ \vdots \\ \boldsymbol{d}_L \end{bmatrix} \begin{matrix} \} \ (n+1)(n+2)/2 \text{ components} \\ \left.\begin{matrix} \\ \\ \\ \end{matrix}\right\} \ r \text{ components} \end{matrix} \tag{38}$$

with $\boldsymbol{a}$, d_0, and $\boldsymbol{d}_i$s defined by (36), (20c), and (23), respectively.

B. Gradient of $H(\omega_1, \omega_2, \boldsymbol{x})$ and Stability Constraints

The gradient of $H(\omega_1, \omega_2, \boldsymbol{x})$ with respect to parameter vector $\boldsymbol{x}$ is an $(\hat{n}+r)$-dimensional vector given by

$$g(\omega_1, \omega_2, \boldsymbol{x}) = \begin{bmatrix} \dfrac{\partial H(\omega_1, \omega_2, \boldsymbol{x})}{\partial \boldsymbol{a}} \\ \dfrac{\partial H(\omega_1, \omega_2, \boldsymbol{x})}{\partial d_0} \\ \dfrac{\partial H(\omega_1, \omega_2, \boldsymbol{x})}{\partial \boldsymbol{d}_1} \\ \vdots \\ \dfrac{\partial H(\omega_1, \omega_2, \boldsymbol{x})}{\partial \boldsymbol{d}_L} \end{bmatrix}. \tag{39}$$

Using (37), the components of $g(\omega_1, \omega_2, \boldsymbol{x})$ can be evaluated as follows:

$$\frac{\partial H(\omega_1, \omega_2, \boldsymbol{x})}{\partial \boldsymbol{a}} = \frac{\boldsymbol{e}(\omega_1, \omega_2)}{d(\omega_1)d(\omega_2)} \tag{40a}$$

$$\frac{\partial H(\omega_1, \omega_2, \boldsymbol{x})}{\partial d_0} = -H(\omega_1, \omega_2, \boldsymbol{x}) \frac{2d_0 v_1(\omega_1)v_1(\omega_2) + v_1(\omega_1) + v_1(\omega_2)}{[1 + d_0 v_1(\omega_1)][1 + d_0 v_1(\omega_2)]} \tag{40b}$$

$$\frac{\partial H(\omega_1, \omega_2, \boldsymbol{x})}{\partial \boldsymbol{d}_i} = -H(\omega_1, \omega_2, \boldsymbol{x}) \cdot \frac{[1 + \boldsymbol{d}_i^T \boldsymbol{v}_2(\omega_2)]\boldsymbol{v}_2(\omega_1) + [1 + \boldsymbol{d}_i^T \boldsymbol{v}_2(\omega_1)]\boldsymbol{v}_2(\omega_2)}{[1 + \boldsymbol{d}_i^T \boldsymbol{v}_2(\omega_1)][1 + \boldsymbol{d}_i^T \boldsymbol{v}_2(\omega_2)]} \tag{40c}$$

for $1 \le i \le L$

where $v_1(\omega)$ and $\boldsymbol{v}_2(\omega)$ are defined in (23). Like the 1-D case, $v_1(\omega_1)$, $v_1(\omega_2)$, $\boldsymbol{v}_2(\omega_1)$ and $\boldsymbol{v}_2(\omega_2)$ are independent of filter coefficients and can be used repeatedly as the optimization iteration proceeds. Furthermore, since $\Omega_2 = \Omega \times \Omega$, both sets $\{v_1(\omega_1), v_1(\omega_2)\}$ and $\{\boldsymbol{v}_2(\omega_1), \boldsymbol{v}_2(\omega_2)\}$ for $(\omega_1, \omega_2) \in \Omega_2$ can be obtained by just evaluating a 1-D data set $\{\boldsymbol{v}_2(\omega), \omega \in \Omega_d\}$ ($\{v_1(\omega)$ is a subset of $\{\boldsymbol{v}_2(\omega)\}$). Therefore, it is worthwhile to prepare and store data $\{\boldsymbol{v}_2(\omega), \omega \in \Omega_d\}$ before the iterations begin. On the other hand, since the dimension of vector $\boldsymbol{e}(\omega_1, \omega_2)$ is usually quite high and $\boldsymbol{e}(\omega_1, \omega_2)$ is defined on a set of dense 2-D grid points, it is more realistic to evaluate $\boldsymbol{e}(\omega_1, \omega_2)$ as the iteration process goes along.

Concerning the robust stability, from (34), it is obvious that the optimized filter is stable with a stability margin $1 - \sqrt{1-\tau}$ if the denominator polynomial $d(z)$ meets the constraint in (32). In other words, the stability constraint for the 2-D IIR filter in (34) remains the same as in the 1-D case.

C. Design Algorithm

For the design methodology outlined in Section IV to fit into the 2-D design scenario, ω, Ω, Ω_d need to be replaced, respectively, by (ω_1, ω_2), Ω_2, and $\Omega_{2d} = \{(\omega_{1i}, \omega_{2i}), 1 \le i \le K\}$, which is a set of grid points placed in the frequency region of interest. Given filter order (n, r) [see (34)], desired frequency response $H_d(\omega_1, \omega_2)$ on Ω_2, and parameter τ, a minimax design of 2-D IIR filter with stability margin $1 - \sqrt{1-\tau}$ can be obtained by carrying out the following steps.

Step 1) Data preparation that includes i) choosing an initial stable design $\boldsymbol{x}_0$ and a set of grid points Ω_{2d}; ii) evaluating $\{\boldsymbol{v}_2(\omega), \omega \in \Omega_d\}$; and iii) setting iteration counter $k = 0$, bound β, and convergence tolerance ε. A straightforward and stable $\boldsymbol{x}_0$ assumes the form

$$\boldsymbol{x}_0 = \begin{bmatrix} \boldsymbol{a}_0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}_{(\hat{n}+r)\times 1}$$

which yields a 2-D filter whose transfer function is given by $H(\omega_1, \omega_2)$ in (35) with $d(\omega_1) = d(\omega_2) = 1$. Vector $\boldsymbol{a}_0$ corresponds to the impulse response of $a(\omega_1, \omega_2)$ in (35b) that approximates $H_d(\omega_1, \omega_2)$. This $a(\omega_1, \omega_2)$ can be readily obtained using an established method [30].

Step 2) At $\boldsymbol{x}_k$, solve the CQP problem in (19) for

$$\boldsymbol{u}_k = \begin{bmatrix} \eta_k \\ \boldsymbol{\delta}_k \end{bmatrix}$$

where $\boldsymbol{G}_k$ and $\boldsymbol{e}_k$ are defined in (14) with $\boldsymbol{g}_k = \boldsymbol{g}(\omega_1, \omega_2, \boldsymbol{x}_k)$ evaluated using (39) and (40); the constraints in (19d) are specified by (32), and $W(\omega_1, \omega_2)$ is a piecewise constant weighting function defined by

$$W(\omega_1, \omega_2) = \begin{cases} 1, & \text{for } (\omega_1, \omega_2) \text{ in passband} \\ w, & \text{for } (\omega_1, \omega_2) \text{ in stopband} \\ 0, & \text{elsewhere.} \end{cases}$$

Step 3) Update the design from $\boldsymbol{x}_k$ to $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{\delta}_k$. If $\|\boldsymbol{\delta}_k\| < \varepsilon$, stop; otherwise, set $k := k+1$, and repeat from Step 2.

D. Design Example

The design concerns a circularly symmetric lowpass filter of order $(n, r) = (12, 8)$, with $\omega_p = 0.5\pi$, $\omega_a = 0.7\pi$, and passband group delay in both ω_1 and ω_2 being eight samples. The number of filter coefficients in this case is $\hat{n} + r = (13 \times 14)/2 + 8 = 99$. The initial point used in the design corresponds to a trivial 2-D IIR filter whose transfer function has $d(z) = 1$ [see (34)], and $a(z_1, z_2)$ is a linear-phase FIR lowpass filter of order 12 that was designed using the singular-value decomposition (SVD) method [30]. The toolbox SeDuMi1.05 [22] was used to implement the proposed algorithm on a 866-MHz Pentium III PC. With $\varepsilon = 10^{-8}$, $w = 1$, $\tau = 0.2$, $\beta = 0.005$, and a total of 1030 grid points placed uniformly in the passband and stopband, it took the algorithm 45 iterations to converge to a design shown in Fig. 6 with performance evaluation given in Table III.

Another design with the same specifications as above but a different initial point obtained using the SVD-balanced approximation method [34] was also carried out. The proposed algo-

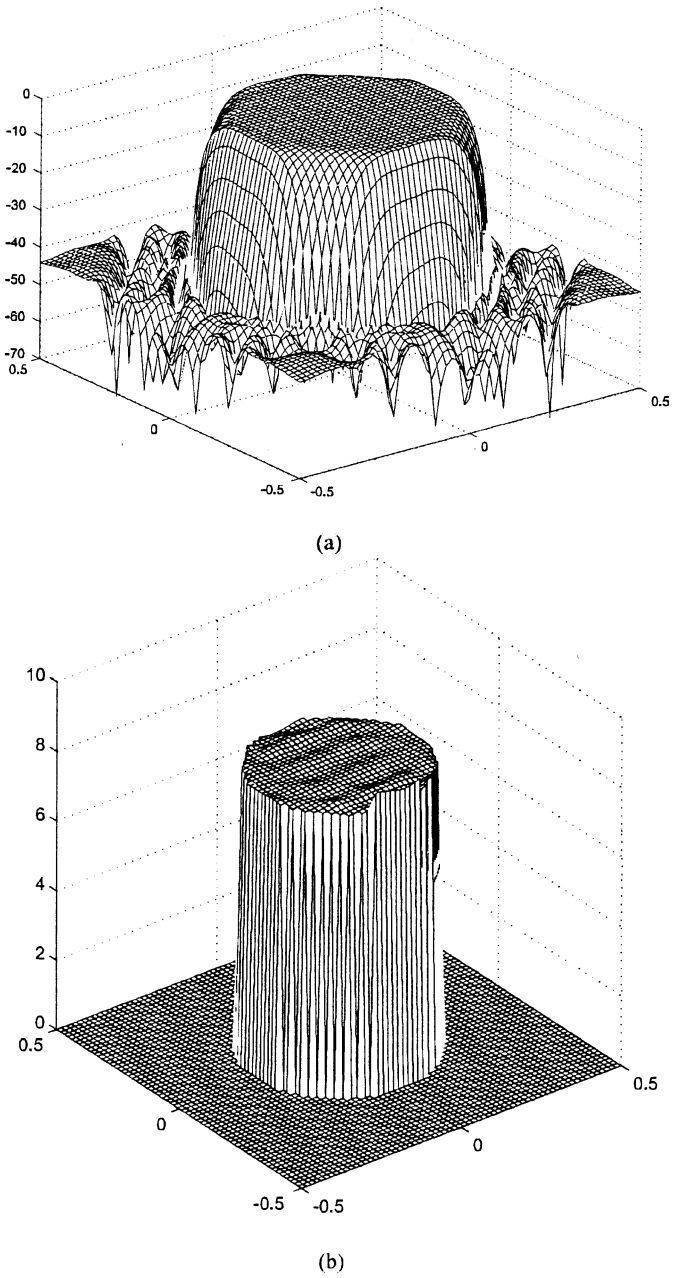


Fig. 6. (a) Amplitude response. (b) Passband group delay of the circularly symmetric lowpass IIR filter.

TABLE III
PERFORMANCE COMPARISON

IIR Filter of Order (n, r)	Design in [16] $n = 12,\ r = 8$	Proposed Design $n = 12,\ r = 8$
maximum amplitude deviation in passband	0.0127	0.0081
minimum stopband attenuation (dB)	37.1798	39.3666
passband group delay (sample)	8	8
average deviation in passband group delay	0.0119	0.0080
maximum magnitude of poles	0.8619	0.8944
CPU time (seconds)	4.85 K	1.62 K
floating point operations	3.98×10^4 M	1.38×10^4 M

rithm in this case converges to the same solution point but with only 21 iterations and 0.65×10^4 Mflops.

In the literature, only a handful of articles were for the minimax design of 2-D filters. Early work in the field includes [31], where only linear-phase FIR filters were considered. Recent work on the minimax design of 2-D IIR filters with guaranteed stability includes [16], where the stability constraint is a linear matrix inequality deduced based on the Lyapunov's stability theory, and the optimization is carried out using SDP rather than CQP. For comparison purposes, the algorithm in [16] was applied to design a lowpass 2-D IIR filter with the same design specifications as described above. The design results are given in Table III. An interpretation of the simulation results is that the less conservative stability constraint in the proposed CQP formulation tends to include more qualified candidates, yielding an improved local minimizer. It is also observed that the CPU time as well as the number of floating-point operations required by the CQP-based algorithm to obtain the design were considerably reduced.

VII. CONCLUSION

We have presented a design methodology in which 1-D and 2-D IIR filters with separable denominators can be synthesized in the minimax sense with prescribed stability margin. The methodology was developed in a rather general setting in which the minimax approximation is accomplished through a sequence of linear updates with each update carried out using conic quadratic programming. As demonstrated by computer simulations, the proposed method yields IIR filters with satisfactory performance.

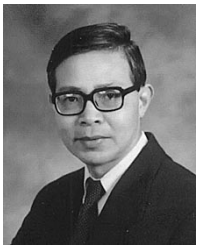
ACKNOWLEDGMENT

The authors are grateful to B. Dumitrescu for providing a preprint of [17].

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. AE-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. AE-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. AE-21, pp. 460–470, Oct. 1973.
[5] A. G. Deczky, "Equiripple and minimax (Chebyshev) approximation 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] M. Ahmadi, A. G. Constantinides, and R. A. King, "Design technique for a class of stable two-dimensional recursive digital filters," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 1976, pp. 145–148.
[8] K. Hirano and J. K. Aggarwal, "Design of two-dimensional recursive digital filters," *IEEE Trans. Circuits Syst.*, vol. CAS-25, pp. 1066–1076, Dec. 1978.
[9] A. T. Chottera and G. A. Jullien, "A linear programming approach to recursive digital filter design with linear phase," *IEEE Trans. Circuits Syst.*, vol. CAS-29, pp. 139–149, Mar. 1982.

[10] T. Hinamoto and S. Maekawa, "Design of two-dimensional recursive digital filters using mirror image polynomials," *IEEE Trans. Circuits Syst.*, vol. CAS-33, pp. 750–758, 1986.
[11] G. Gu and B. A. Shenoi, "A novel approach to the synthesis of recursive digital filters with linear phase," *IEEE Trans. Circuits Syst.*, vol. 38, pp. 602–612, June 1991.
[12] 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.
[13] W.-S. Lu, "Design of recursive digital filters with prescribed stability margin: A parameterized approach," *IEEE Trans. Circuits Syst.*, vol. 48, pp. 1289–1298, Sept. 1998.
[14] M. C. Lang, "Least-squares design of IIR filters with prescribed magnitude and phase response and a pole radius constraint," *IEEE Trans. Signal Processing*, vol. 48, pp. 3109–3121, Nov. 2000.
[15] W.-S. Lu, "Design of stable minimax IIR digital filters using semidefinite programming," in *Proc. Int. Symp. Circuits Syst.*, vol. 1, May 2000, pp. 355–358.
[16] W.-S. Lu and A. Antoniou, "Minimax design of 2-D IIR digital filters using sequential semidefinite programming," in *Proc. Int. Symp. Circuits Syst.*, May 2002.
[17] B. Dumitrescu, "On convex stability domain and optimization of IIR filters," in *Proc. EUSIPCO*, vol. 2, Sept. 2002, pp. 191–194, submitted for publication.
[18] M. S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret, "Applications of second-order cone programming," *Linear Algebra Applications*, vol. 248, pp. 193–228, Nov. 1998.
[19] J. O. Coleman and D. P. Scholnik, "Design of nonlinear-phase FIR filters with second-order cone programming," in *Proc. Midwest Symp. Circuits Syst.*, Las Cruces, NM, Aug. 1999.
[20] A. Ben-Tal and A. Nemirovski, *Lectures on Modern Convex Optimization*. Philadelphia, PA: SIAM, 2001.
[21] L. Vandenberghe and S. Boyd, "Semidefinite programming," *SIAM Rev.*, vol. 38, pp. 49–95, Mar. 1996.
[22] J. F. Sturm, "Using SeDuMi1.02, a MATLAB toolbox for optimization over symmetric cones," *Optim. Methods Softw.*, vol. 11–12, pp. 625–653, 1999.
[23] R. H. Tütüncü, K. C. Toh, and M. J. Todd, *SDPT3—A MATLAB Software Package for Semidefinite-Quadratic-Linear Programming, version 3.0*. Natick, MA: MathWorks, Aug. 2001.
[24] P. Gahinet, A. Nemirovski, A. J. Laub, and M. Chilali, *Manual of LMI Control Toolbox*. Natick, MA: MathWorks, 1995.
[25] Y. E. Nesterov and A. Nemirovski, *Interior-Point Polynomial Methods in Convex Programming*. Philadelphia, PA: SIAM, 1994.
[26] Y. E. Nesterov and M. J. Todd, "Self-scaled barriers and interior-point methods for convex programming," *Math. Oper. Res.*, vol. 22, pp. 1–42, 1997.
[27] J. F. Sturm, *Primal-Dual Interior-Point Approach to Semidefinite Programming*. Amsterdam, Netherlands: Tinbergen Inst., 1997, vol. 156.
[28] J. E. Dennis, Jr. and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Philadelphia, PA: SIAM, 1996.
[29] R. K. Rajan and M. N. S. Swamy, "Quadrantal symmetry associated with two-dimensional digital transfer functions," *IEEE Trans. Circuits Syst.*, vol. CAS-25, pp. 340–343, June 1983.
[30] W.-S. Lu and A. Antoniou, *Two-Dimensional Digital Filters*. New York: Marcel Dekker, 1992.
[31] C. Charalambous, "The performance of an algorithm of minimax design of two-dimensional linear phase FIR digital filters," *IEEE Trans. Circuits Syst.*, vol. CAS-32, pp. 1016–1028, Oct. 1985.
[32] Z.-Q. Luo, J. F. Sturm, and S. Zhang, "Conic convex programming and self-dual embedding," *Optim. Methods Softw.*, vol. 14, no. 3, pp. 169–218, 2000.
[33] H. Kimura and Y. Honoki, "Balanced approximation of digital FIR filter with linear phase characteristic," in *Proc. Int. Symp. Circuits Syst.*, 1985, pp. 283–286.
[34] W.-S. Lu, H.-P. Wang, and A. Antoniou, "Design of two-dimensional digital filters using singular-value decomposition and balanced approximation method," *IEEE Trans. Signal Processing*, vol. 39, pp. 2253–2262, Oct. 1991.



Wu-Sheng Lu (F'99) received his undergraduate education in mathematics from Fudan University, Shanghai, China, from 1959 to 1964 and the M.S. degree in electrical engineering and Ph.D. degree in control science from University of Minnesota, Minneapolis, in 1983, and 1984, respectively.

He was a post-doctoral fellow at the University of Victoria, Victoria, BC, Canada, in 1985 and a visiting assistant professor at University of Minnesota in 1986. Since 1987, he has been with University of Victoria, where he is currently a Professor. His teaching and research interests are in the areas of digital signal processing and application of optimization methods. He is the coauthor, with A. Antoniou, of *Two-Dimensional Digital Filters* (New York: Marcel Dekker, 1992). He was an Associate Editor of the *Canadian Journal of Electrical and Computer Engineering* in 1989 and the Editor of the same journal from 1990 to 1992. He is presently an Associate Editor for the *International Journal of Multidimensional Systems and Signal Processing*.

Dr. Lu served as an Associate Editor for IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II from 1993 to 1995 and for IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS I from 1999 to 2001. He is a Fellow of the Engineering Institute of Canada.

Takao Hinamoto (M'77–SM'84–F'01) received the B.E. degree from Okayama University, Okayama, Japan, in 1969, the M.E. degree from Kobe University, Kobe, Japan, in 1971, and the Dr. Eng. degree from Osaka University, Osaka, Japan, in 1977, all in electrical engineering.

From 1972 to 1988, he was with the Faculty of Engineering, Kobe University. From 1979 to 1981, he was on leave from Kobe University as visiting member of staff in the Department of Electrical Engineering, Queen's University, Kingston, ON, Canada. From 1988 to 1991, he was Professor of electronic circuits with the Faculty of Engineering, Tottori University, Tottori, Japan. Since January 1992, he has been Professor of electronic control with the Department of Electrical Engineering, Hiroshima University, Hiroshima, Japan. His research interests include digital signal processing, system theory, and control engineering. He has published more than 270 papers in these areas and is the co-editor and co-author of *Two-Dimensional Signal and Image Processing* (Tokyo, Japan: SICE, 1996).

Dr. Hinamoto served as an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II from 1993 to 1995 and presently serves as an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS I. He also served as Chair of the 12th Digital Signal Processing (DSP) Symposium held in Hiroshima in November 1997, sponsored by the DSP Technical Committee of IEICE. He was the Guest Editor of the special section on Digital Signal Processing in the August 1998 issue of the *IEICE Transactions on Fundamentals*. Since 1995, he has been a member of the steering committee of the IEEE Midwest Symposium on Circuits and Systems and, since 1998, a member of the Digital Signal Processing Technical Committee of the IEEE Circuits and Systems Society. He served as a member of the Technical Program Committee for ISCAS'99. From 1993 to 2000, he served as a senator or member of the Board of Directors in the Society of Instrument and Control Engineers (SICE), and from 1999 to 2001, he was Chair of the Chugoku Chapter of SICE. He played a leading role in establishing the Hiroshima Section of IEEE and served as the Interim Chair of the section. He is a recipient of the IEEE Third Millennium Medal.