Optimal Design of IIR Digital Filters with Robust Stability Using Conic Quadratic Programming

Wu-Sheng Lu
Dept. of Elec. and Comp. Eng.
University of Victoria, Canada
E-mail: wslu@ece.uvic.ca

Takao Hinamoto
Graduate School of Engineering
Hiroshima University, Japan
E-mail: hinamoto@ecl.sys.hiroshima-u.ac.jp

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. Also included in the formulation is a second-order cone condition on the magnitude of each update that ensures the validity of a key linear approximation used in the design and eliminates a line-search step. Collectively, these features lead to improved designs relative to several established methods.

1. 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].

A major problem encountered in the design of IIR filters is stability. A 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 [3][4], Rouché's condition on denominator perturbations [5], iterative Lyapunov inequality constraints [6][7], or a general positive realness constraint on denominator perturbations [8]. 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 IIR digital filters. The design method has several features: (i) The design is acomplished 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 [12][13]. (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 [10][11] 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.

2. Preliminaries

2.1 Stability Triangle of Second-Order Systems

Consider the transfer function of a second-order discrete-time system whose denominator is given by $d(z) = z^2 + d_1 z + d_2$. The system is stable if and only if

$$\boldsymbol{C}_2 \boldsymbol{d} + \hat{\boldsymbol{e}} > \boldsymbol{0} \qquad \text{(1a)}$$

where

$$\boldsymbol{C}_2 = \begin{bmatrix} 1 & 1 \\ -1 & 1 \\ 0 & -1 \end{bmatrix}, \ \boldsymbol{d} = \begin{bmatrix} d_1 \\ d_2 \end{bmatrix}, \ \hat{\boldsymbol{e}} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \qquad \text{(1b)}$$

The constraints in (1) are *linear* w.r.t. d_1 and d_2, and characterize the triangle in the $(d_1,\ d_2)$-space shown in Fig. 1.

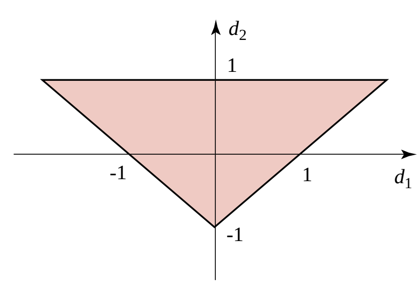

Figure 1. Stability triangle

For the sake of robust stability, we consider a triangle in (d_1, d_2)-space that is strictly inside the stability trangle in

Fig. 1. See Fig. 2 for the illustration. The region endosed with the internal triangle is characterized by

$$\boldsymbol{C}_2\boldsymbol{d} + (1-\tau)\hat{\boldsymbol{e}} \geq \boldsymbol{0} \tag{2}$$

2.2. Conic Quadratic Programming

Conic quadratic programming, which is sometimes called the second-order cone programming [9], is a subclass of convex programming problems where a linear function is minimized subject to a set of second-order cone constraints [9][11]:

$$\text{minimize} \quad \boldsymbol{f}^T\boldsymbol{x} \tag{3a}$$

$$\text{subject to:} \quad \|\boldsymbol{A}_i\boldsymbol{x} + \boldsymbol{b}_i\| \leq \boldsymbol{c}_i^T\boldsymbol{x} + h_i, i = 1, \ldots, N \tag{3b}$$

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 (3b) 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.,

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

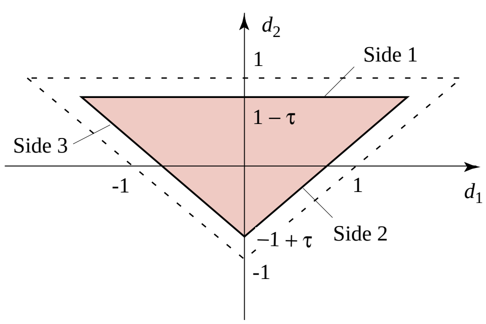

Figure 2. An internal stability triangle

3. A general design method using linear CQP updates

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

$$\underset{\boldsymbol{x}}{\text{minimize}}\{\underset{\omega\in\Omega}{\text{maximize}}\ W(\omega)|H(\omega,\,\boldsymbol{x}) - H_d(\omega)|\} \tag{4a}$$

$$\text{subject to:} \quad H(\omega,\,\boldsymbol{x}) \text{ stable} \tag{4b}$$

If η denotes an upper bound of $W(\omega)|H(\omega,\,\boldsymbol{x}) - H_d(\omega)|$ on Ω, then the problem in (4) can be converted into

$$\text{minimize} \quad \eta \tag{5a}$$

$$\text{subject to:} \quad W(\omega)|H(\omega,\,\boldsymbol{x}) - H_d(\omega)| \leq \eta \quad \omega \in \Omega \tag{5b}$$

$$H(\omega,\,\boldsymbol{x}) \text{ stable} \tag{5c}$$

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

$$H(\omega,\,\boldsymbol{x}_k + \boldsymbol{\delta}) \approx H(\omega,\,\boldsymbol{x}_k) + \boldsymbol{g}_k^T(\omega)\boldsymbol{\delta} \tag{6}$$

provided that

$$\|\boldsymbol{\delta}\| \text{ is small} \tag{7}$$

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 (7), we have

$$|H(\omega,\,\boldsymbol{x}) - H_d(\omega)| \approx |\boldsymbol{g}_k^T(\omega)\boldsymbol{\delta} + [H(\omega,\,\boldsymbol{x}_k) - H_d(\omega)]|$$

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

$$H(\omega,\,\boldsymbol{x}) = H_r(\omega,\,\boldsymbol{x}) + jH_i(\omega,\,\boldsymbol{x}) \tag{8a}$$

$$H_d(\omega) = H_{rd}(\omega) + jH_{id}(\omega) \tag{8b}$$

$$\boldsymbol{g}_k(\omega) = \boldsymbol{g}_{rk}(\omega) + j\boldsymbol{g}_{ik}(\omega) \tag{8c}$$

It follows that

$$W(\omega)|H(\omega,\,\boldsymbol{x}) - H_d(\omega)| \approx \|\boldsymbol{G}_k(\omega)\boldsymbol{\delta} + \boldsymbol{e}_k(\omega)\| \tag{9}$$

where

$$\begin{aligned} \boldsymbol{G}_k(\omega) &= W(\omega)\begin{bmatrix} \boldsymbol{g}_{rk}^T(\omega) \\ = \boldsymbol{g}_{ik}^T(\omega) \end{bmatrix} \\ \boldsymbol{e}_k(\omega) &= W(\omega)\begin{bmatrix} e_{rk}(\omega) \\ e_{ik}(\omega) \end{bmatrix} \\ e_{rk}(\omega) &= H_r(\omega,\,\boldsymbol{x}_k) - H_{rd}(\omega) \\ e_{ik}(\omega) &= H_i(\omega,\,\boldsymbol{x}_k) - H_{id}(\omega) \end{aligned}$$

In the light of (5b), (7), and (9), we see that an approximate solution in the kth iteration can be obtained by solving the constrained optimization problem

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

$$\text{subject to:} \quad \|\boldsymbol{G}_k(\omega)\boldsymbol{\delta} + \boldsymbol{e}_k(\omega)\| \leq \eta \quad \omega \in \Omega \tag{10b}$$

$$\|\boldsymbol{\delta}\| \leq \beta \tag{10c}$$

$$H(\omega,\,\boldsymbol{x}_k + \boldsymbol{\delta}) \quad \text{stable} \tag{10d}$$

where β is a prescribed bound to control the magnitude of $\boldsymbol{\delta}$. Once a solution of (10), 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 (10a) and (10b) as an additional design variable and define an augmented parameter vector

$$\boldsymbol{u} = \begin{bmatrix} \eta \\ \boldsymbol{\delta} \end{bmatrix} \tag{11}$$

then the problem in (10) can be expressed as

$$\begin{aligned} &\text{minimize} \quad \boldsymbol{c}^T\boldsymbol{u} && \text{(12a)}\\ &\text{subject to:} \quad \|\hat{\boldsymbol{G}}_k(\omega)\boldsymbol{u} + \boldsymbol{e}_k(\omega)\| \le \boldsymbol{c}^T\boldsymbol{u} \text{ for } \omega \in \Omega_d && \text{(12b)}\\ &\qquad \|\hat{\boldsymbol{I}}\boldsymbol{u}\| \le \beta && \text{(12c)}\\ &\qquad H(\omega,\, \boldsymbol{x}_k + \boldsymbol{\delta}) \ \text{ stable} && \text{(12d)} \end{aligned}$$

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.

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 (12d) can be characterized by a set of linear inequality constrains as

$$\boldsymbol{C}\boldsymbol{u} + \boldsymbol{h} \ge \boldsymbol{0} \qquad (13)$$

(see Sec. 4.3 for the structure of matrix $\boldsymbol{C}$ and vector $\boldsymbol{h}$).

Suppose matrix $\boldsymbol{C}$ has m rows, then (13) 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 (12) becomes

$$\begin{aligned} &\text{minimize} \quad \boldsymbol{c}^T\boldsymbol{u} && \text{(14a)}\\ &\text{subject to:} \quad \|\hat{\boldsymbol{G}}_k(\omega_i)\boldsymbol{u} + \boldsymbol{e}_k(\omega_i)\| \le \boldsymbol{c}^T\boldsymbol{u} \text{ for } 1 \le i \le K && \text{(14b)}\\ &\qquad \|\hat{\boldsymbol{I}}\boldsymbol{u}\| \le \beta && \text{(14c)}\\ &\qquad \boldsymbol{c}_i^T\boldsymbol{u} + h_i \ge 0 \qquad \text{for } 1 \le i \le m && \text{(14d)} \end{aligned}$$

On comparing the problem in (14) with that in (3), it is evident that problem (14) is a CQP problem with $p+1$ design variables, $K+1$ second-order cone constraints, and m linear constraints.

Several interior-point methods for CQP have been developed in the past, see for example [14]–[16], and [9]. Lucid exposition of the subject can be found in [11].

It should also be pointed out that although problem (14) is merely an *approximation* of (5), as the iteration continues and the local minimizer gets closer, the increment vector $\boldsymbol{\delta}$ obtained by solving (14) 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.

4. Design of 1-D IIR Filters

4.1. The Design Problem

Consider the transfer function of an IIR digital filter

$$H(z) = \frac{a(z)}{z^{n-r}d(z)} \qquad (15a)$$

where

$$a(z) = \sum_{i=0}^{n} a_i z^{n-i} \qquad (15b)$$

$d(z)$ is a polynomial of order r expressed as product of 2nd-order sections (and a first-order section if r is odd):

$$d(z) = \begin{cases} \displaystyle\prod_{i=1}^{r/2}(z^2 + d_{i1}z + d_{i2}) & \text{if } r \text{ even}\\ \displaystyle(z + d_o)\prod_{i=1}^{(r-1)/2}(z^2 + d_{i1}z + d_{i2}) & \text{if } r \text{ odd} \end{cases} \qquad (15c)$$

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 [5]. The design problem at hand is to determine the coefficients of $H(z)$ in (15) that solves the minimax optimization problem

$$\underset{\boldsymbol{x}}{\text{minimize}}[\underset{\omega\in\Omega}{\text{maximize}}\ W(\omega)|H(\omega,\, \boldsymbol{x}) - H_d(\omega)|] \qquad (16a)$$

$$\text{subject to: } d(z) \ne 0 \qquad \text{for } |z| > \sqrt{1-\tau} \qquad (16b)$$

where the filter coefficients form vector $\boldsymbol{x} = [a_0 \ \ \ldots a_n \ \ d_0 \ \ d_{11} \ \ d_{12} \ \ \ldots \ \ d_{L1} \ \ d_{L2}]^T$ with L representing the $L = r/2$ if r even and $(r-1)/2$ if odd, and $W(\omega) \ge 0$ 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

$$H(\omega,\, \boldsymbol{x}) = \frac{a(\omega)}{d(\omega)} \qquad (17)$$

$$\begin{aligned} a(\omega) &= \boldsymbol{a}^T\boldsymbol{v}(\omega),\ \boldsymbol{a} = [a_0 \ a_1 \ \ldots \ a_n]^T\\ \boldsymbol{v}(\omega) &= \boldsymbol{c}(\omega) - j\boldsymbol{s}(\omega)\\ \boldsymbol{c}(\omega) &= [1 \ \ \cos\omega \ \ \cdots \ \ \cos n\omega]^T\\ \boldsymbol{s}(\omega) &= [0 \ \ \sin\omega \ \ \cdots \ \ \sin n\omega]^T\\ d(\omega) &= \begin{cases} \displaystyle\prod_{i=1}^{L}[1 + \boldsymbol{d}_i^T\boldsymbol{v}_2(\omega)] & \text{if } r \text{ even}\\ \displaystyle[1 + d_0 v_1(\omega)]\prod_{i=1}^{L}[1 + \boldsymbol{d}_i^T\boldsymbol{v}_2(\omega)] & \text{if } r \text{ odd} \end{cases}\\ v_1(\omega) &= \cos\omega - j\sin\omega\\ \boldsymbol{d}_i &= \begin{bmatrix} d_{i1}\\ d_{i2} \end{bmatrix},\ \boldsymbol{v}_2(\omega) = \begin{bmatrix} \cos\omega\\ \cos 2\omega \end{bmatrix} - j\begin{bmatrix} \sin\omega\\ \sin 2\omega \end{bmatrix} \end{aligned}$$

The constraint in (16b) characterizes the requirement of robust stability that the pole radius of the filter be $\sqrt{1-\tau}$. On comparing (16) with (4), 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 (14).

4.2. 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 (17) as

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{a} \\ \boldsymbol{d} \end{bmatrix} = \begin{bmatrix} \boldsymbol{a} \\ d_0 \\ \boldsymbol{d}_1 \\ \vdots \\ \boldsymbol{d}_L \end{bmatrix} \begin{matrix} \} \; n+1 \text{ components} \\ \left.\begin{matrix} \\ \\ \\ \\ \end{matrix}\right\} r \text{ components} \end{matrix} \tag{18}$$

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 (17), the gradient of $H(\omega,\, \boldsymbol{x})$ with respect to $\boldsymbol{x}$ is evaluated as

$$\boldsymbol{g}(\omega,\, \boldsymbol{x}) = \begin{bmatrix} \frac{\partial H(\omega, \boldsymbol{x})}{\partial \boldsymbol{a}} \\ \frac{\partial H(\omega, \boldsymbol{x})}{\partial d_0} \\ \frac{\partial H(\omega, \boldsymbol{x})}{\partial \boldsymbol{d}_1} \\ \vdots \\ \frac{\partial H(\omega, \boldsymbol{x})}{\partial \boldsymbol{d}_L} \end{bmatrix} \tag{19}$$

with

$$\frac{\partial H(\omega, \boldsymbol{x})}{\partial \boldsymbol{a}} = \frac{\boldsymbol{v}(\omega)}{d(\omega)} \tag{20a}$$

$$\frac{\partial H(\omega, \boldsymbol{x})}{\partial d_0} = -H(\omega,\, \boldsymbol{x}) \frac{v_1(\omega)}{1 + d_0 v_1(\omega)} \tag{20b}$$

$$\frac{\partial H(\omega, \boldsymbol{x})}{\partial \boldsymbol{d}_i} = -H(\omega,\, \boldsymbol{x}) \frac{\boldsymbol{v}_2(\omega)}{1 + \boldsymbol{d}_i^T \boldsymbol{v}_2(\omega)} \tag{20c}$$

4.3. Constraints for Robust Stability

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

$$\boldsymbol{x}_k + \boldsymbol{\delta} = \begin{bmatrix} \boldsymbol{a} + \boldsymbol{\delta}_a \\ \boldsymbol{d} + \boldsymbol{\delta}_d \end{bmatrix} \tag{21}$$

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

$$\boldsymbol{d} + \boldsymbol{\delta}_d = \begin{bmatrix} d_0 + \delta_0 \\ \boldsymbol{d}_1 + \boldsymbol{\delta}_1 \\ \vdots \\ \boldsymbol{d}_L + \boldsymbol{\delta}_L \end{bmatrix} \tag{22}$$

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

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

i.e.,

$$\begin{bmatrix} 1 \\ -1 \end{bmatrix} (d_0 + \delta_0) + (1 - \tau) \begin{bmatrix} 1 \\ 1 \end{bmatrix} \geq \boldsymbol{0} \tag{23a}$$

Each vector $\boldsymbol{d}_i + \boldsymbol{\delta}_i$ is connected to a 2nd-order section in $d(z)$ whose robust stability is satisfied if (2) is imposed upon, i.e.,

$$\boldsymbol{C}_2(\boldsymbol{d}_i + \boldsymbol{\delta}_i) + (1 - \tau)\hat{\boldsymbol{e}} \geq \boldsymbol{0} \qquad \text{for } 1 \leq i \leq L \tag{23b}$$

where $\boldsymbol{C}_2$ and $\hat{\boldsymbol{e}}$ are defined in (1b). Therefore, $\boldsymbol{x}_k + \boldsymbol{\delta}$ in (21) represents an IIR filter with stability margin $1 - \sqrt{1-\tau}$ if

$$\hat{\boldsymbol{C}}(\boldsymbol{d} + \boldsymbol{\delta}_d) + (1 - \tau)\boldsymbol{e} \geq \boldsymbol{0} \tag{24}$$

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

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

with $\boldsymbol{c}_1 = [1 \;\; -1]^T$ (if r is even, then the top-left $\boldsymbol{c}_1$ in $\hat{\boldsymbol{C}}$ does not present and $m = 3L$). Now if we augment matrix $\hat{\boldsymbol{C}}$ in (24) with $n+1$ columns of zeros on the left and replace $\boldsymbol{d} + \boldsymbol{\delta}_d$ there by $\boldsymbol{x}_k + \boldsymbol{\delta}$, then (24) becomes

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

i.e.,

$$[\boldsymbol{0} \;\; \hat{\boldsymbol{C}}]\boldsymbol{\delta} + \boldsymbol{h} \geq \boldsymbol{0} \tag{25a}$$

where

$$\boldsymbol{h} = [\underbrace{\boldsymbol{0}}_{(n+1) \text{ columns}} \;\; \hat{\boldsymbol{C}}]\boldsymbol{x}_k + (1 - \tau)\boldsymbol{e} \tag{25b}$$

Finally, by augmenting the matrix in (25a) with one more zero column on the left and replacing vector $\boldsymbol{\delta}$ there by $\boldsymbol{u}$ (defined in (11)), the stability constraint in (25) becomes

$$\boldsymbol{C}\boldsymbol{u} + \boldsymbol{h} \geq \boldsymbol{0} \tag{26}$$

where

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

Equivalently, (26) can be expressed as m linear inequality constraints as seen in (14d) where $\boldsymbol{c}_i$ denotes the ith column of matrix $\boldsymbol{C}^T$ and h_i is the ith component of $\boldsymbol{h}$.

4.4. A 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 [2], 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 Table 1 and Fig. 3 (dash-dotted curves). 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 [12] 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 [17] 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 seconds of CPU time. 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 $\boldsymbol{x}_0^{(1)}$. The performance of the IIR filter designed are evaluated in terms of Error of frequency response in passband, passband magnitude ripple, stopband attenuation, average deviation in passband group delay, and maximum magnitude of the poles, and is illustrated in Fig. 3 and Table 1. From Fig. 3 and Table 1, considerable performance improvement over the Deczky filter were observed.

Table 1. 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

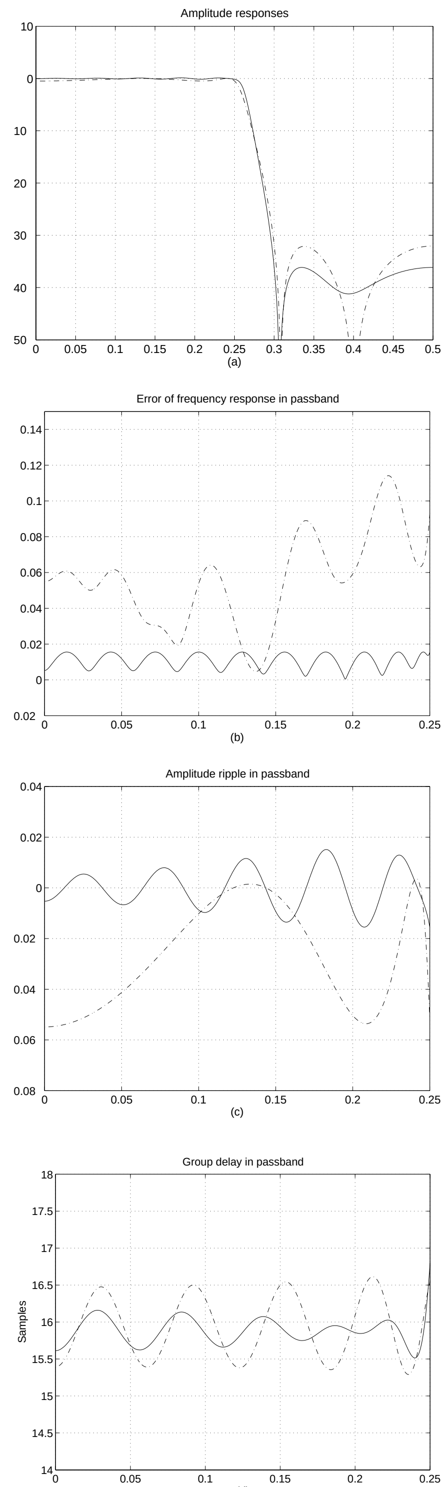

Figure 3. (a) Amplitude responses, (b) error of frequency response in passband , (c) passband amplitude responses, and (d) passband group delays of the proposed design (solid curves) the Deczky filter (dash-dotted curves).

References

[1] A. Antoniou, *Digital Filters: Analysis and Design*, 2nd ed. New York: McGraw-Hill, 1993.

[2] A. G. Deczky, "Equiripple and minimax (Chebyshev) approximation for recursive digital filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 22, pp. 98–111, Apr. 1974.

[3] 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.

[4] 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.

[5] 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.

[6] W.-S. Lu, "Design of stable minimax IIR digital filters using semidefinit programming," *Proc. Int. Symp. Circuits Syst.*, May 2000.

[7] W.-S. Lu and A. Antoniou, "Minimax design of 2-D IIR digital filters using sequential semidefinite programming," *Proc. Int. Symp. Circuits Syst.*, May 2002.

[8] B. Dumitrescu, "On convex stability domain and optimization of IIR filters," submitted to *EUSIPCO'2002*.

[9] M. S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret, "Applications of second-order cone programming," *Linear Algebra and Applications*, vol. 248, pp. 193–228, Nov. 1998.

[10] J. O. Coleman and D. P. Scholnik, "Design of nonlinear-phase FIR filters with second-order cone programming," *Proc. Midwest Symp. Circuits Syst.*, Las Cruces, NM, Aug. 1999.

[11] A. Ben-Tal and A. Nemirovski, *Lectures on Modern Convex Optimization*, SIAM, Philadelphia, 2001.

[12] J. F. Sturm, "Using SeDuMi1.02, a MATLAB toolbox for optimization over symmetric cones," *Optimization Methods and Software*, vol. 11–12, pp. 625–653, 1999.

[13] R. H. Tütüncü, K. C. Toh, and M. J. Todd, "*SDPT3—A MATLAB Software Package for Semidefinite-Quadratic-Linear Programming, version 3.0*," Aug. 2001.

[14] Y. E. Nesterov and A. Nemirovski, *Interior-Point Polynomial Methods in Convex Programming*, SIAM, Philadelphia, 1994.

[15] Y. E. Nesterov and M. J. Todd, "Self-scaled barriers and interior-point methods for convex programming," *Math. Operations Research*, vol. 22, pp. 1–42, 1997.

[16] J. F. Sturm, *Primal-Dual Interior-Point Approach to Semidefinite Programming*, vol 156 in Tinbergen Institute Research Series, Amsterdam, Netherlands, 1997.

[17] H. Kimura and Y. Honoki, "Balanced approximation of digital FIR filter with linear phase characteristic," *Int. Symp. Circuits Syst.*, pp. 283-286, 1985.