ISCAS 2000 - IEEE International Symposium on Circuits and Systems, May 28-31, 2000, Geneva, Switzerland

# DESIGN OF STABLE 2-D IIR DIGITAL FILTERS USING ITERATIVE SEMIDEFINITE PROGRAMMING

*W.-S. Lu*

Dept. of Electrical and Computer Engineering
University of Victoria
Victoria, BC, Canada V8W 3P6

## ABSTRACT

Semidefinite programming (SDP) has recently attracted a great deal of research interest. Among other things, the optimization tool was proven to be applicable to design various types of FIR digital filters. This paper describes an attempt on extending the SDP approach to 2-D IIR filters. It is shown that a stable 2-D IIR filter design in the minimax sense can be formulated as an iterative SDP problem. Stability constraints are expressed as linear matrix inequalities which fit nicely into the SDP-based design setting. Unlike the 1-D case, an alternating iteration scheme is used to deal with parameter nonlinearity encountered in the separable denominator polynomials. A design example is presented to illustrate the proposed method.

## 1. INTRODUCTION

Recursive digital filters offer improved selectivity, computation efficiency and reduced system delay compared to what can be achieved by nonrecursive digital filters [1]–[5]. Like the 1-D IIR filters, the major drawbacks of 2-D IIR designs are that linear phase response can be achieved only approximately and the designs must handle stability problem which does not exist in the FIR case. Several recent methods formulated 1-D IIR designs as constrained optimization problems [4], [6]–[9]. In [9], the 1-D IIR design problem was formulated as a semidefinite programming (SDP) problem. SDP is a relatively new interior-point optimization methodology that has attracted a great deal of research interest in the past several years [10][11].

In this paper, we describe a SDP-based method for the design of stable 2-D IIR filters. The proposed method is applicable to design recursive 2-D filters that approximate given arbitrary frequency responses. For the sake of notation and description simplicity, however, here we restrict ourselves to the class of filters with separable denominators, which covers virtually all useful 2-D filters. An algorithmic feature of the method proposed here that distinguishes itself from the SDP-based method in [9] is that it employs an alternating iteration scheme to deal with the parameter nonlinearity encountered in the separable denominator polynomial. Stability constraints are expressed as linear matrix inequalities (LMI's) which fit nicely into the SDP-based design setting. The proposed method is illustrated by a design example.

## 2. SDP AND STABILITY CONSTRAINTS

### 2.1. SDP

There are several ways a SDP problem can be formulated. For the purpose of filter design, the following formulation turns out to be of convenience:

$$\text{minimize} \quad \mathbf{c}^T\mathbf{x} \tag{1a}$$

$$\text{subject to} \quad \mathbf{F}(\mathbf{x}) \succeq \mathbf{0} \tag{1b}$$

$$\mathbf{F}(\mathbf{x}) = \mathbf{F}_0 + \sum_{i=1}^{n} x_i\mathbf{F}_i \tag{1c}$$

In (1), $\mathbf{x} = [x_1 \ \ldots \ x_n]^T$ is the variable vector, $\mathbf{c} \in R^{n\times 1}$, $\mathbf{F}_i \in R^{n\times n}$ for $0 \le i \le n$ are constant matrices with $\mathbf{F}_i$ symmetric, and $\mathbf{F}(\mathbf{x}) \succeq \mathbf{0}$ denotes that the constraint matrix $\mathbf{F}(\mathbf{x})$ is positive semidefinite. Note that the constraint matrix is *affine* with respect to $\mathbf{x}$. SDP includes both linear and convex quadratic programming as special cases. More importantly, many interior-point methods, which have proven efficient for linear programming, have recently been extended to SDP [10][11]. Efficient and user-friendly software implementations of various SDP algorithms are available. In particular, we mention the LMI Control Toolbox, which works with MATLAB, as an excellent implementation tool [12].

### 2.2. Stability Constraints

Consider a separable denominator polynomial of order $(r_1, r_2)$

$$d(z_1, z_2) = f(z_1)g(z_2) \tag{2}$$

where

$$f(z_1) = z_1^{r_1} + f_1 z_1^{r_1-1} + \cdots + f_{r_1} \tag{3}$$

$$g(z_2) = z_2^{r_2} + g_1 z_2^{r_2-1} + \cdots + g_{r_2} \tag{4}$$

Polynomial $d(z_1, z_2)$ is said to be stable if $d(z_1, z_2) \neq 0$ in the region $\{(z_1, z_2) : |z_1| \ge 1, \ |z_2| \ge 1\}$. A 1-D polynomial $f(z_1)$ is said to be stable, if the zeros of $f(z_1)$ are in the region $\{z_1 : |z_1| < 1\}$. Clearly $d(z_1, z_2)$ in (1) is stable if and only if both $f(z_1)$ and $g(z_2)$ are stable. Defining the canonical matrices

$$\mathbf{D}_f = \begin{bmatrix} -f_1 & -f_2 & \cdots & -f_{r_1} \\ 1 & & & 0 \\ & \ddots & & \vdots \\ & & 1 & 0 \end{bmatrix} \tag{5}$$

and

$$\mathbf{D}_g = \begin{bmatrix} -g_1 & -g_2 & \cdots & -g_{r_1} \\ 1 & & & 0 \\ & \ddots & & \vdots \\ & & 1 & 0 \end{bmatrix} \tag{6}$$

then the zeros of $f(z_1)$ and $g(z_2)$ are the eigenvalues of $\mathbf{D}_f$ and $\mathbf{D}_g$, respectively. We say a matrix $\mathbf{D}$ is stable if the modulus of its eigenvalues are strictly less than 1. Hence an irreducible transfer function $b(z_1, z_2)/f(z_1)g(z_2)$ represents a stable IIR 2-D filter if and only if both $\mathbf{D}_f$ and $\mathbf{D}_g$ are stable. From the well known Lyapunov theory [13], it follows that $\mathbf{D}_f$ and $\mathbf{D}_g$ are stable if and only if there exist positive definite matrices $\mathbf{P}$ and $\mathbf{Q}$ such that

$$\mathbf{P} - \mathbf{D}_f^T \mathbf{P} \mathbf{D}_f \succ \mathbf{0} \tag{7}$$

and

$$\mathbf{Q} - \mathbf{D}_g^T \mathbf{Q} \mathbf{D}_g \succ \mathbf{0} \tag{8}$$

It can be readily verified that (7) and (8) hold if and only if

$$\begin{bmatrix} \mathbf{P}^{-1} & \mathbf{D}_f \\ \mathbf{D}_f^T & \mathbf{P} \end{bmatrix} \succ \mathbf{0} \tag{9}$$

and

$$\begin{bmatrix} \mathbf{Q}^{-1} & \mathbf{D}_g \\ \mathbf{D}_g^T & \mathbf{Q} \end{bmatrix} \succ \mathbf{0} \tag{10}$$

respectively. As will be seen in the next section, the LMI constraints in (9) and (10) are of convenience to use in an SDP-based design, because parameter matrices $\mathbf{D}_f$ and $\mathbf{D}_g$ appear *affinely*.

## 3. DESIGN FORMULATION AND ALGORITHM

### 3.1. Design Formulation

Let the transfer function of a 2-D IIR digital filter be denoted by

$$H(z_1, z_2) = \frac{b(z_1, z_2)}{d(z_1, z_2)} \tag{11}$$

where

$$b(z_1, z_2) = \sum_{i=0}^{n} \sum_{k=0}^{m} b_{ik} z_1^{n-i} z_2^{m-k} \tag{12}$$

$$d(z_1, z_2) = f(z_1) g(z_2) \tag{13}$$

$$f(z_1) = z_1^{n-r_1} \sum_{i=0}^{r_1} f_i z_1^{r_1-i} \quad \text{with } f_0 = 1 \tag{14}$$

$$g(z_1) = z_2^{m-r_2} \sum_{i=0}^{r_2} g_i z_2^{r_2-i} \quad \text{with } g_0 = 1 \tag{15}$$

and $r_1$ and $r_2$ are integers with $0 \le r_1 \le n$ and $0 \le r_2 \le m$. This form of $d(z_1, z_2)$ is convenient to preserve a certain number of poles at the origin as it might be beneficial for the design of several types of digital filters [4][8]. The frequency response of the filter can now be written as

$$H(\omega_1, \omega_2) = \frac{b(\omega_1, \omega_2)}{d(\omega_1, \omega_2)} \tag{16}$$

where

$$b(\omega_1, \omega_2) = \sum_{i=0}^{n} \sum_{k=0}^{m} b_{ik} e^{-(i\omega_1 + k\omega_2)j} \tag{17}$$

$$d(\omega_1, \omega_2) = \left( \sum_{i=0}^{r_1} f_i e^{-ji\omega_1} \right) \left( \sum_{i=0}^{r_2} g_i e^{-ji\omega_2} \right) \tag{18}$$

The design problem here is to find a stable $H(z_1, z_2)$ that best approximates a given 2-D frequency response $H_d(\omega_1, \omega_2)$ in the minimax sense. Namely, $H(z_1, z_2)$ solves the following constrained optimization problem

$$\begin{aligned} &\text{minimize} && \underset{-\pi \le \omega_1, \omega_2 \le \pi}{\text{maximize}} \; e(\omega_1, \omega_2) && \text{(19a)} \\ &\text{subject to} && f(z_1) \ne 0 \quad \text{for } |z_1| \ge 1 && \text{(19b)} \\ &&& g(z_2) \ne 0 \quad \text{for } |z_2| \ge 1 && \text{(19c)} \end{aligned}$$

where

$$e(\omega_1, \omega_2) = W(\omega_1, \omega_2) |H(\omega_1, \omega_2) - H_d(\omega_1, \omega_2)|$$

and $W(\omega_1, \omega_2) \ge 0$ is a weighting function.

### 3.2. Assumptions and Notation

For the sake of description simplicity we assume that $n = m$ and $r_1 = r_2 = r$. With several minor modifications in the notation, the design algorithm to be described can be applied to the cases where $n \ne m$ and $r_1 \ne r_2$. The following notation will be adopted in the rest of the paper:

$$\begin{aligned} \mathbf{b} &= [b_{00}\, b_{10} \cdots b_{n0}\, b_{01} \cdots b_{nn}]^T \\ \mathbf{f} &= [f_1\, f_2 \cdots f_r]^T \\ \mathbf{g} &= [g_1\, g_2 \cdots g_r]^T \\ \mathbf{c}(\omega_1, \omega_2) &= \\ &[1\, \cos\omega_1 \cdots \cos n\omega_1\, \cos\omega_2 \cdots \cos(n\omega_1 + n\omega_2)]^T \\ \mathbf{s}(\omega_1, \omega_2) &= \\ &[0\, \sin\omega_1 \cdots \sin n\omega_1\, \sin\omega_2 \cdots \sin(n\omega_1 + n\omega_2)]^T \\ \mathbf{c}_1(\omega_i) &= [\cos\omega_i \cdots \cos r\omega_i]^T \quad \text{for } i = 1, 2 \\ \mathbf{s}_1(\omega_i) &= [\sin\omega_i \cdots \sin r\omega_i]^T \quad \text{for } i = 1, 2 \\ H_d(\omega_1, \omega_2) &= H_r(\omega_1, \omega_2) - jH_i(\omega_1, \omega_2) \end{aligned}$$

where $H_r(\omega_1, \omega_2)$ and $-H_i(\omega_1, \omega_2)$ are the real and imaginary parts of $H_d(\omega_1, \omega_2)$, respectively.

### 3.3. An Iterative SDP Formulation

First, we re-formulate the minimax problem in (19) as

$$\begin{aligned} &\text{minimize} && \delta && \text{(20a)} \\ &\text{subject to} && e^2(\omega_1, \omega_2) \le \delta && \text{(20b)} \\ &&& f(z_1) \ne 0 \quad \text{for } |z_1| \ge 1 && \text{(20c)} \\ &&& g(z_2) \ne 0 \quad \text{for } |z_2| \ge 1 && \text{(20d)} \end{aligned}$$

where the upper bound $\delta$ will be treated as an auxiliary design variable. Next we write the constraint in (20b) as

$$e^2(\omega_1, \omega_2) = \frac{W^2}{|f|^2 |g|^2} |b - fgH_d|^2 \le \delta \tag{21}$$

where for the simplicity of writing the dependence of $W, b, f, g,$ and $H_d$ on $\omega_1$, $\omega_2$ have been omitted. Note that the term $fgH_d$ is *nonlinear* in terms of the design variables. This makes it difficult to convert the problem into a SDP problem because the square of the term $fgH_d$ in magnitude is no longer quadratic. A remedy for this technical difficulty is to use an alternating iteration scheme as described below.

Suppose an initial $b_0(z_1, z_2)$ and a stable pair $f_0(z_1)$, $g_0(z_2)$ has been chosen. For an integer $k \ge 1$, (21) suggests to first solve the following constrained problem for $b(z_1, z_2)$ and a stable $f(z_1)$:

$$\text{minimize} \quad \delta \tag{22a}$$

$$\text{subject to} \quad \frac{W^2}{|f_{k-1}|^2 |g_{k-1}|^2} |b - f g_{k-1} H_d|^2 \leq \delta \tag{22b}$$

$$f(z_1) \neq 0 \qquad \text{for } |z_1| \geq 1 \tag{22c}$$

Let the minimizing $b(z_1, z_2)$ and $f(z_1)$ be denoted by $b_k(z_1, z_2)$ and $f_k(z_1)$, respectively. Next one seeks to find a stable $g_k(z_2)$ and a (new) $b_k(z_1, z_2)$ that solves the constrained problem

$$\text{minimize} \quad \delta \tag{23a}$$

$$\text{subject to} \quad \frac{W^2}{|f_{k-1}|^2 |g_{k-1}|^2} |b - f_{k-1} g H_d|^2 \leq \delta \tag{23b}$$

$$g(z_2) \neq 0 \qquad \text{for } |z_2| \geq 1 \tag{23c}$$

The iterations described above are similar in spirit to the Steiglitz-McBride (SM) scheme which finds applications in system identification and adaptive filtering [14]. The difference between (22), (23) and the SM scheme is that the SM scheme iterates a least-squares objective function while each of (22) and (23) involves an iterative constraint in a minimax problem.

By straightforward manipulations it can be verified that the constraint in (22b) can be expressed as a parameterized LMI constraint

$$\mathbf{\Phi}(\delta, \mathbf{x}, \omega_1, \omega_2) \succeq \mathbf{0} \tag{24}$$

where

$$\mathbf{\Phi}(\delta, \mathbf{x}, \omega_1, \omega_2) = \begin{bmatrix} \delta & a_1 & a_2 \\ a_1 & 1 & 0 \\ a_2 & 0 & 1 \end{bmatrix}$$

with

$$\begin{aligned}
a_1 &= \mathbf{x}^T \mathbf{c}_k - H_{a_1}, \; a_2 = \mathbf{x}^T \mathbf{s}_k - H_{a_2} \\
\mathbf{x} &= \begin{bmatrix} \mathbf{b} \\ \mathbf{f} \end{bmatrix}, \; \mathbf{c}_k = \begin{bmatrix} \mathbf{c}_w \\ \mathbf{u}_v \end{bmatrix}, \; \mathbf{s}_k = \begin{bmatrix} \mathbf{s}_w \\ \mathbf{v}_w \end{bmatrix} \\
w_k &= \frac{W(\omega_1, \omega_2)}{|f_{k-1}(\omega_1) g_{k-1}(\omega_2)|} \\
\mathbf{c}_w &= w_k \mathbf{c}(\omega_1, \omega_2), \; \mathbf{s}_w = w_k \mathbf{s}(\omega_1, \omega_2) \\
\mathbf{u}_w &= w_k [H_r (e_{k-1} \mathbf{s}_1 - d_{k-1} \mathbf{c}_1) + H_i (d_{k-1} \mathbf{s}_1 + e_{k-1} \mathbf{c}_1)] \\
\mathbf{v}_w &= w_k [H_r (d_{k-1} \mathbf{s}_1 + e_{k-1} \mathbf{c}_1) + H_i (d_{k-1} \mathbf{c}_1 - e_{k-1} \mathbf{s}_1)] \\
H_{a_1} &= w_k [H_r d_{k-1} - H_i e_{k-1}] \\
H_{a_2} &= w_k [-H_r e_{k-1} - H_i d_{k-1}] \\
d_{k-1} &= 1 + \mathbf{g}_{k-1}^T \mathbf{c}_1 \\
e_{k-1} &= \mathbf{g}_{k-1}^T \mathbf{s}_1
\end{aligned}$$

Note that matrix $\mathbf{\Phi}(\delta, \mathbf{x}, \omega_1, \omega_2)$ depends on design parameters $\delta$ and $\mathbf{x}$ *affinely*, and the stability of $f_{k-1}(z)$ and $g_{k-1}(z)$ obtained from the $(k-1)$th iteration assures a well-defined weighting function $w_k$.

Concerning the stability constraint in (22c), from Sec. 2.2 it follows that for a stable $f_{k-1}(z)$ there exists a $\mathbf{P}_{k-1} \succ \mathbf{0}$ that satisfies the Lyapunov equation

$$\mathbf{P}_{k-1} - \mathbf{D}_{f_{k-1}}^T \mathbf{P}_{k-1} \mathbf{D}_{f_{k-1}} = \mathbf{I} \tag{25}$$

where $\mathbf{D}_{f_{k-1}}$ is the canonical matrix (5) with $-\mathbf{f}_{k-1}$ in its first row, and $\mathbf{I}$ is the $r_1 \times r_1$ identify matrix. From Sec. 2.2, a natural stability constraint for $f_k(z)$ is

$$\mathbf{S}_k(\mathbf{x}) = \begin{bmatrix} \mathbf{P}_{k-1}^{-1} - \tau \mathbf{I} & \mathbf{D}_f \\ \mathbf{D}_f^T & \mathbf{P}_{k-1} - \tau \mathbf{I} \end{bmatrix} \succeq \mathbf{0} \tag{26}$$

where $\mathbf{D}_f$ is defined by (5) and $\tau > 0$ is a scalar introduced to control the stability margin of $f_k(z_1)$. It is observed that $\mathbf{S}_k(\mathbf{x})$ depends on $\mathbf{D}_f$ and hence $\mathbf{x}$ affinely, therefore (26) is an LMI. Also notice that the positive define matrix $\mathbf{P}_{k-1}$ in (26) is obtained from (25), hence it is "constrained" by $\mathbf{D}_{f_{k-1}}$. As a result, (26) is a sufficient (but not necessary) constraint for the stability of $f_k(z_1)$. However, if the sequence of iterates $\{f_k(z)\}$ converges as the iterations continue, then the matrix sequence $\{\mathbf{D}_{f_k}\}$ will converge as well. Since the existence of $\mathbf{P}_{k-1} \succ \mathbf{0}$ in (25) is a necessary and sufficient condition for the stability of $f_{k-1}(z_1)$, the LMI constraint in (26) gets less and less restrictive as the algorithm proceeds. We consider this a desirable feature of the proposed algorithm.

At the $k$th iteration the constrained optimization problem in (22) can now be formulated as

$$\text{minimize} \quad \hat{\mathbf{c}}^T \hat{\mathbf{x}} \tag{27a}$$

$$\text{subject to} \quad \begin{bmatrix} \mathbf{\Phi}_k(\hat{\mathbf{x}}) & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_k(\mathbf{x}) \end{bmatrix} \succeq \mathbf{0} \tag{27b}$$

where $\hat{\mathbf{x}}$ is the augmented variable defined by

$$\hat{\mathbf{x}} = \begin{bmatrix} \delta \\ \mathbf{x} \end{bmatrix} = \begin{bmatrix} \delta \\ \mathbf{b} \\ \mathbf{f} \end{bmatrix}, \; \hat{\mathbf{c}} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{27c}$$

and $\mathbf{\Phi}_k(\hat{\mathbf{x}}) \succeq \mathbf{0}$ is a discrete implementation of (24) on a set of frequencies $\{(\omega_1^{(l)}, \omega_2^{(m)}), \text{ for } 1 \leq l \leq L, 1 \leq m \leq M\}$ in a frequency region of interest:

$$\mathbf{\Phi}_k(\hat{\mathbf{x}}) = \text{diag}\{\mathbf{\Phi}(\hat{\mathbf{x}}, \omega_1^{(1)}, \omega_2^{(1)}), \cdots, \mathbf{\Phi}(\hat{\mathbf{x}}, \omega_1^{(L)}, \omega_2^{(M)})\} \tag{27d}$$

Since both $\mathbf{\Phi}_k(\hat{\mathbf{x}})$ and $\mathbf{S}_k(\mathbf{x})$ depend on $\hat{\mathbf{x}}$ affinely, (27) is an SDP problem as defined in (1).

By a similar analysis, it can be verified that the companion optimization problem in (23) at the $k$th iteration can also be formulated as an SDP problem:

$$\text{minimize} \quad \hat{\mathbf{c}}^T \hat{\mathbf{y}} \tag{28a}$$

$$\text{subject to} \quad \begin{bmatrix} \mathbf{\Psi}_k(\hat{\mathbf{y}}) & \mathbf{0} \\ \mathbf{0} & \mathbf{T}_k(\mathbf{y}) \end{bmatrix} \succeq \mathbf{0} \tag{28b}$$

where

$$\hat{\mathbf{y}} = \begin{bmatrix} \delta \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} \delta \\ \mathbf{b} \\ \mathbf{g} \end{bmatrix}, \; \hat{\mathbf{c}} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{28c}$$

and matrices $\mathbf{\Psi}_k(\hat{\mathbf{y}})$ and $\mathbf{T}_k(\mathbf{y})$ are defined in a manner similar to that of $\mathbf{\Phi}_k(\hat{\mathbf{x}})$ and $\mathbf{S}_k(\mathbf{x})$, respectively.

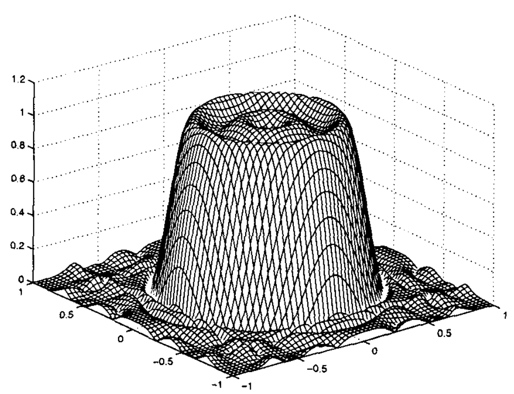
### 3.4. The Algorithm

Given a desired frequency response $H_d(\omega_1, \omega_2)$, a weighting function $W(\omega_1, \omega_2)$, and filter order $(n, r)$, one chooses a pair of convenient initial points $\mathbf{x}_0 = [\mathbf{b}_0^T \; \mathbf{f}_0^T]^T$ $\mathbf{y}_0 = [\mathbf{b}_0^T \; \mathbf{g}_0^T]^T$ where

$\mathbf{f}_0 = \mathbf{g}_0 = \mathbf{0}$ and $\mathbf{b}_0$ is obtained by designing a 2-D FIR filter to approximate $H_d(\omega_1, \omega_2)$ using a routine design method. Next one solves the SDP problems in (27) and then (28) for $k = 1$, and evaluates $\varepsilon_k = ||\mathbf{x}_k - \mathbf{x}_{k-1}|| + ||\mathbf{y}_k - \mathbf{y}_{k-1}||$. If $\varepsilon_k$ is less then a prescribed tolerance $\varepsilon$, then the $\mathbf{b}_k, \mathbf{g}_k$ from $\mathbf{y}_k$ and $\mathbf{f}_k$ from $\mathbf{x}_k$ are deemed as the optimal solution for the design problem. Otherwise the algorithm proceeds by solving (27) and then (28) for $k = 2$, etc. The algorithm is illustrated in the next section by a design example.

## 4. A DESIGN EXAMPLE

The proposed algorithm was used to design several 2-D IIR filters, including circular symmetric and diamond-shaped lowpass, highpass, bandpass, and bandstop filters. As an example, we design an $(n, r) = (12, 8)$ circular symmetric, lowpass IIR filter with passband edge $= 0.5\pi$, stopband edge $= 0.7\pi$, and linear phase response in the passband with group delay in both directions being 6 samples. With $n = 12$, $r = 8$, $\omega_p = 0.5\pi$, $\omega_a = 0.7\pi$, $\varepsilon = 0.05$, $\tau = 10^{-12}$, $W(\omega_1, \omega_2) \equiv 1$ in the union of the passband and stopband regions and $W(\omega_1, \omega_2) \equiv 0$ elsewhere, it took the proposed algorithm 7 iterations to converge to a solution whose amplitude and phase (in passband) responses are depicted in Fig. 1. The maximum modulus of the poles of $b(z_1, z_2)/f(z_1)g(z_2)$ was 0.8859. The maximum amplitude deviation in passband and stopband are 0.0578 and 0.1454, respectively. The proposed method was favorably compared with other methods [5] for designing 2-D IIR filters in terms of design efficiency and filter's performance. The comparison results are omitted here due to space limitation. Finally, we remark that a fairly low group delay can be achieved by the proposed IIR design compared to that of a linear-phase 2-D FIR filter with comparable approximation accuracy.

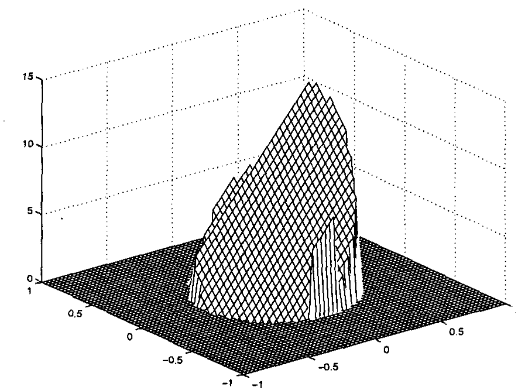



(a)

## 5. REFERENCES

[1] A. Antoniou, *Digital Filters: Analysis, Design, and Applications,* 2nd ed., Mc-Graw-Hill, 1993.

[2] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*, Prentice-Hall, 1989.




(b)

Figure 1: (a) Amplitude responses, (b) Phse response in passband.

[3] A. G. Deczky, "Synthesis of recursive digital filters using the minimum $p$-error criterion", *IEEE Trans. Audio and Electroacoustics*, vol. 20, pp. 257-263, 1972.

[4] M. Lang, "Algorithms for the constrained design of digital filters with arbitrary magnitude and phase responses", Doctoral dissertation, Vienna Univ. of Technology, June 1999.

[5] W.-S. Lu and A. Antoniou, *Two-Dimensional Digital Filters*, Marcel Dekker, 1992.

[6] 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, March 1982.

[7] 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.

[8] M. Lang, "Weighted least squares IIR filter design with arbitrary magnitude and phase responses and specified stability margin", *IEEE DFSP'98*, pp. 82-86, Victoria, Canada, June 1998.

[9] W.-S. Lu, "Design of stable minimax IIR digital filters using semidefinite programming", submitted to *ISCAS'2000*.

[10] Y. Nesterov and A. Nemirovskii, *Interior Point Polynomial Methods in Convex Programming*, SIAM, Philadelphia, 1994.

[11] L. Vandenberghe and S. Boyd, "Semidefinite programming", *SIAM Review*, vol. 38, pp. 49-95, March 1996.

[12] P. Gahinet, A. Nemirovski, A. J. Laub, and M. Chilali, *Manual of LMI Control Toolbox*, The MathWorks Inc, Natick, MA, 1995.

[13] T. Kailath, *Linear Systems*, Prentice-Hall, 1981.

[14] P. A. Regalia, *Adaptive IIR Filtering in Signal Processing and Control*, Marcel Dekker, 1995.