

Model Reduction of 2-D IIR Filters

Lahcène Mitiche¹, Amel Baha Houda Adamou-Mitiche¹, Omar Kacem¹, Vasil Sima²

¹Science and Technology Department, University of Djelfa, Djelfa, Algeria; ²National Institute for Research & Development in Informatics, Bucharest, Romania.

Email: l_mitiche@yahoo.fr, amelmitiche@yahoo.fr, kacem.omar@yahoo.fr, vsima@ici.ro

Received June 24th, 2012; revised September 11th, 2012; accepted September 23rd, 2012

ABSTRACT

The work presented in this paper concerns with analysis and synthesis of the two-dimensional Infinite Impulse Response (IIR) filters based on model order reduction. The synthesis is performed with two methods, the Prony's method (Prony modified) and Iterative method, in the spatial domain, and with the method of Semi-Definite iterative Programming (SDP), in the frequency domain. After synthesis, we make an order reduction of the filter model by the Quasi-Gramians method.

Keywords: Stability; Recursively Computable; Linear Phase; State Space; Balanced Realization; Model Reduction

1. Introduction

The fields of two-dimensional digital signal processing and digital image processing have maintained tremendous vitality over the past four decades and there is a clear indication that this trend will continue. Advances in hardware technology provide capabilities in signal processing chips and microprocessors which were previously associated with mainframe computers. These advances allow sophisticated signal and image processing algorithms to be implemented in real time at a substantially reduced cost. New applications continue to be found and existing applications continue to expand in such diverse areas as communications, consumer electronics, medicine, defense, robotics, and geophysics.

At a conceptual level, there is a great deal of similarity between one-dimensional signal processing and two-dimensional signal processing. In one-dimensional signal processing, the concepts discussed are filtering, Fourier transform, discrete Fourier transform, fast Fourier transform algorithms, and so on. In two-dimensional signal processing, we again are concerned with the same concepts.

At a more detailed level, however, considerable differences exist between one-dimensional and two-dimensional signal processing. For example, one major difference is the amount of data involved in typical applications. In speech processing, we have 10.000 data points to process in a second. However, in video processing, we would have 7.5 million data points to process per second. Another example is the absence of the fundamental theorem of algebra for two-dimensional polynomials.

One-dimensional polynomials can be factored as a product of lower-order polynomials. An important structure for realizing a one-dimensional digital filter is the cascade structure. In this case, the z -transform of the digital filter's impulse response is factored as a product of lower-order polynomials and the realizations of these lower-order factors are cascaded. The z -transform of a two-dimensional digital filters impulse response cannot, in general, be factored as a product of lower-order polynomials and the cascade structure therefore is not a general structure for a two-dimensional digital filter realization [1]. Another consequence of the nonfactorability of a two-dimensional polynomial is the difficulty associated with issues related to system stability. In a one-dimensional system, the pole locations can be determined easily, and an unstable system can be stabilized without affecting the magnitude response by simple manipulation of pole locations. In a two-dimensional system, since poles are surfaces rather than points and there is no fundamental theorem of algebra, it is extremely difficult to determine the pole locations [2].

The paper presented here is related particularly to the synthesis of the two-dimensional infinite impulse response filters using model order reduction. There are two main parts of the paper. In the first part, the synthesis is done, both in the spatial domain, with two methods, the Prony's method (modified Prony) and Iterative method, and in the frequency domain, with iterative Semi-Definite Programming (SDP). In the second part of this paper, we make an order reduction of the synthesized (designed) filter using the Quasi-Gramians approach.

2. Designing 2D-IIR Filters

An 2D-IIR filter with an arbitrary impulse response $h(n_1, n_2)$ cannot be realized since computing each output sample requires a too large number of arithmetic operations [1,2]. As a result, in addition to requiring $h(n_1, n_2)$ to be real and stable, we require $h(n_1, n_2)$ to have a rational z-transform corresponding to a recursively computable system.

2.1. The Design Problem

The problem of IIR filter design is to determine a rational and stable function $H(z_1, z_2)$ with a wedge support output mask that meets a given design specification. In other words, we wish to determine a stable computational procedure that is recursively computable and meets a design specification.

However, a given rational system function $H(z_1, z_2)$ can lead to many different computational procedures [1]. To make the relationship unique, we will adopt a convention in expressing $H(z_1, z_2)$. Specifically, we will assume that $a(0, 0)$ is always 1, so $H(z_1, z_2)$ will then be in the form

$$H(z_1, z_2) = \frac{\sum_{(k_1, k_2) \in R_b} b(k_1, k_2) z_1^{-k_1} z_2^{-k_2}}{1 + \sum_{(k_1, k_2) \in R_a - (0,0)} a(k_1, k_2) z_1^{-k_1} z_2^{-k_2}} \quad (1)$$

where $R_a - (0, 0)$ represents the support region of $a(k_1, k_2)$ without the origin $(0, 0)$, and R_b represents the support region of $b(k_1, k_2)$.

The unique computational procedure corresponding to (1) is then given by

$$y(n_1, n_2) \leftarrow \sum_{(k_1, k_2) \in R_a - (0,0)} a(k_1, k_2) y(n_1 - k_1, n_2 - k_2) + \sum_{(k_1, k_2) \in R_b} b(k_1, k_2) x(n_1 - k_1, n_2 - k_2) \quad (2)$$

where the sequences $a(k_1, k_2)$ and $b(k_1, k_2)$ are the filter coefficients.

The first step in the IIR filter design is usually an initial determination of R_a and R_b , the support regions of $a(k_1, k_2)$ and $b(k_1, k_2)$. If we determine the filter coefficients by attempting to approximate some desired impulse response $h_d(n_1, n_2)$ in the spatial domain, we will want to choose R_a and R_b such that $h(n_1, n_2)$ will have at least approximately the same support region as $h_d(n_1, n_2)$.

Another consideration is related to the filter specification parameters. In the low-pass filter design, for example, small δ_p , δ_s , (filter templates) and transition region will generally require a larger number of filter coefficients. It is often difficult to determine the number of filter coefficients required to meet a given filter speci-

fication for a particular design algorithm, and an iterative procedure may become necessary [1].

One major difference between IIR and FIR filters is related to stability. A FIR filter is always stable as long as $h(n_1, n_2)$ is bounded (finite) for all (n_1, n_2) , so stability is never an issue. With an IIR filter, however, ensuring stability is a major task. One approach to designing a stable IIR filter is to impose a special structure on $H(z_1, z_2)$ such that testing the stability and stabilizing an unstable filter become relatively easy tasks. Such an approach, however, tends to impose a severe constraint on the design algorithm or to highly restrict the class of filters that can be designed [1]. For example, if $H(z_1, z_2)$ has a separable denominator polynomial of the form $A_1(z_1)A_2(z_2)$, testing the stability and stabilizing an unstable $H(z_1, z_2)$ without affecting the magnitude response is a 1-D problem. However, the class of filters that can be designed with a separable denominator polynomial without a significant increase in the number of coefficients in the numerator polynomial of $H(z_1, z_2)$ is restricted. An alternative approach is to design a filter without considering the stability issue, and then test the stability of the resulting filter and attempt to stabilize it if it proves unstable. However, testing stability and stabilizing an unstable filter are not easy problems.

In the 1-D case, there are two standard approaches to designing IIR filters. One is to design the filter from an analog system function. And the other is to design directly in the discrete domain. The first approach is typically much simpler and more useful than the second: using an elliptic analog filter's system function and a bilinear transformation. Unfortunately, this approach is not useful in the 2-D case. In the 1-D case, this approach exploits the availability of many simple methods to design 1-D analog filters. Such simple methods do not exist for the design of 2-D analog filters. The second approach is the design of IIR filter directly in the discrete domain, can be classified into two categories. The first is the spatial domain design technique, where filters are designed by using an error criterion in the spatial domain. The second is the frequency domain design technique, where filters are designed by using an error criterion in the frequency domain. Therefore, the weighted Chebyshev error criterion, also known as the min-max error criterion, is a natural choice for designing IIR filters. An error criterion of this type, however, leads to a highly nonlinear problem [1,2].

2.2. The Stability Problem

In the 1-D case, testing the stability of a causal system whose system function is given by $H(z) = 1/A(z)$ is quite straightforward. Since a 1-D polynomial $A(z)$ can always be factored straightforwardly as a product of

first-order polynomials, we can easily determine the poles of $H(z)$. The stability of the causal system is equivalent to having all the poles inside the unit circle. The above approach cannot be used in testing the stability of a 2-D first quadrant support system. That approach requires the specific location of all poles to be determined. Partly because a 2-D polynomial $A(z_1, z_2)$ cannot in general be factored as a product of lower-order polynomials, it is extremely difficult to determine all the pole surfaces of $H(z_1, z_2) = 1/A(z_1, z_2)$, and the approach based on explicit determination of all pole surfaces has not led to successful practical procedures for testing the system stability [1,2].

2.3. Spatial Domain Design

The input often used in IIR filter design is $\delta(n_1, n_2)$, and the desired impulse response, assumed given, is denoted by $h_d(n_1, n_2)$. Spatial domain design can be viewed as a system identification problem. Suppose we have an unknown system that we wish to model with a rational system function $H(z_1, z_2)$. One approach to estimating the system model parameters (filter coefficients $a(k_1, k_2)$ and $b(k_1, k_2)$ in our case), is to require the impulse response of the designed system to be as close as possible in some sense to $h_d(n_1, n_2)$. The error criterion used in the filter design is

$$\text{Error} = \sum_{(n_1, n_2) \in R_e} \sum e^2(n_1, n_2), \tag{3a}$$

where

$$e(n_1, n_2) = h_d(n_1, n_2) - h(n_1, n_2), \tag{3b}$$

and R_e is the support region of the error sequence. Ideally, R_e coincides with all values of (n_1, n_2) . The mean square error in (3) is chosen because it is used widely in a number of system identification problems and some variation of it serves as the basis for a number of simple methods developed to estimate the system parameters.

Minimizing the error in (3) with respect to $a(k_1, k_2)$ and $b(k_1, k_2)$ is a nonlinear problem. An approach is to slightly modify the error in (3), so that the resulting algorithm leads to closed form solutions that require solving only sets of linear equations [1].

Consider a computational procedure given by

$$y(n_1, n_2) \leftarrow - \sum_{(k_1, k_2) \in R_a} \sum a(k_1, k_2) y(n_1 - k_1, n_2 - k_2) + \sum_{(k_1, k_2) \in R_b} \sum b(k_1, k_2) x(n_1 - k_1, n_2 - k_2) \tag{4}$$

We will assume that there are p unknown values of $a(k_1, k_2)$ and $q + 1$ unknown values of $b(k_1, k_2)$, and thus a total of $N = p + q + 1$ filter coefficients to be determined for a given pair (n_1, n_2) .

Replacing $x(n_1, n_2)$ with $\delta(n_1, n_2)$ and $y(n_1, n_2)$ with $h_d(n_1, n_2)$ in (4) and noting that

$$\sum_{(k_1, k_2) \in R_b} \sum b(k_1, k_2) \delta(n_1 - k_1, n_2 - k_2)$$

is $b(n_1, n_2)$, we have

$$h_d(n_1, n_2) \leftarrow - \sum_{(k_1, k_2) \in R_a} \sum a(k_1, k_2) h_d(n_1 - k_1, n_2 - k_2) + b(n_1, n_2) \tag{5}$$

Since we wish to approximate $h_d(n_1, n_2)$ as well as we can with $h(n_1, n_2)$, it is reasonable to define an error sequence $e_M(n_1, n_2)$ as the difference between the left-hand and right-hand side expressions of (5)

$$e_M(n_1, n_2) = h_d(n_1, n_2) + \sum_{(k_1, k_2) \in R_a} \sum a(k_1, k_2) h_d(n_1 - k_1, n_2 - k_2) - b(n_1, n_2). \tag{6}$$

It is clear that $e_M(n_1, n_2)$ in (6) is not the same as $e(n_1, n_2)$ in (3b). The subscript M in $e_M(n_1, n_2)$ is used to emphasize that $e_M(n_1, n_2)$ is a modification of $e(n_1, n_2)$. Minimizing $e_M(n_1, n_2)$ with respect to the unknown coefficients $a(n_1, n_2)$ and $b(n_1, n_2)$ is a linear problem.

a) Prony's method

In Prony's method, the error expression minimized is

$$\text{Error} = \sum_{n_1 = -\infty}^{\infty} \sum_{n_2 = -\infty}^{\infty} e_M^2(n_1, n_2), \tag{7}$$

where $e_M(n_1, n_2)$ is given by (6).

The error in (7) is a quadratic form of the unknown parameters $a(n_1, n_2)$ and $b(n_1, n_2)$. Careful observation of the error in (7) shows that can be solved by first solving p linear equations for $a(n_1, n_2)$ and then solving $q + 1$ linear equations for $b(n_1, n_2)$. It is useful to rewrite (7) as

$$\text{Error} = E_1 + E_2, \tag{8a}$$

where

$$E_1 = \sum_{(n_1, n_2) \in R_b} \sum e_M^2(n_1, n_2), \tag{8b}$$

and

$$E_2 = \sum_{(n_1, n_2) \notin R_b} \sum e_M^2(n_1, n_2). \tag{8c}$$

The expression E_1 in (8b) consists of $q + 1$ terms, and E_2 in (8c) consists of a large number of terms. Minimizing E_2 in (8) with respect to $a(n_1, n_2)$ results in p linear equations for p unknowns given by

$$\begin{aligned} & \sum_{(k_1, k_2) \in R_a - (0,0)} \sum a(k_1, k_2) r(k_1, k_2; l_1, l_2) \\ &= -r(0, 0; l_1, l_2), \quad (l_1, l_2) \in R_a - (0, 0) \end{aligned} \quad (9a)$$

where

$$\begin{aligned} & r(k_1, k_2; l_1, l_2) \\ &= \sum_{(n_1, n_2) \in R_b} \sum h_d(n_1 - k_1, n_2 - k_2) \\ & \times h_d(n_1 - l_1, n_2 - l_2). \end{aligned} \quad (9b)$$

Once $a(n_1, n_2)$ is determined, we can minimize the error in (8a) with respect to $b(n_1, n_2)$.

Since Prony's method attempts to reduce the total square error, the resulting filter is likely to be stable [2].

b) Iterative algorithm

The Iterative Algorithm is an extension of a 1-D system identification method developed by Steiglitz and McBride [1,2].

From (6), $e(n_1, n_2) = h_d(n_1, n_2) - h(n_1, n_2)$ is related to $e_M(n_1, n_2)$ by

$$e_M(n_1, n_2) = a(n_1, n_2) * e(n_1, n_2) \quad (10)$$

Equation (10) can be rewritten as

$$e(n_1, n_2) = v(n_1, n_2) * e_M(n_1, n_2) \quad (11)$$

The sequence $v(n_1, n_2)$ is the inverse of $a(n_1, n_2)$. From (6) and (11)

$$\begin{aligned} e(n_1, n_2) &= v(n_1, n_2) * e_M(n_1, n_2) \\ &= v(n_1, n_2) * (a(n_1, n_2) * h_d(n_1, n_2) - b(n_1, n_2)) \end{aligned} \quad (12)$$

From (12), if $v(n_1, n_2)$ is somehow given, then $e(n_1, n_2)$ is linear in both $a(n_1, n_2)$ and $b(n_1, n_2)$, so minimization of $\sum_{n_1} \sum_{n_2} e^2(n_1, n_2)$ with respect to $a(n_1, n_2)$ and $b(n_1, n_2)$ is a linear problem.

Algorithm

Step 1: We start with an initial estimate of $a(n_1, n_2)$, obtained following a method such as Prony.

Step 2: Obtain $v(n_1, n_2)$ from $a(n_1, n_2)$.

Step 3: Minimize $\sum_{n_1} \sum_{n_2} e^2(n_1, n_2)$ with respect to $a(n_1, n_2)$ and $b(n_1, n_2)$ by solving a set of linear equations.

Step 4: We now have a new estimate of $a(n_1, n_2)$, and the process continues until we get a desired $a(n_1, n_2)$ and $b(n_1, n_2)$.

c) Zero-phase filter design

One characteristic of a zero-phase filter is its tendency to preserve the shape of the signal component in the pass band region of the filter. In applications such as speech processing, the zero-phase (or linear phase) characteristic of a filter is not very critical. The human auditory system responds to short time spectral magnitude characteristics,

so the shape of a speech waveform can sometimes change drastically without the human listener's being able to distinguish it from the original. In image processing, the linear phase characteristic appears to be more important. Our visual world consists of lines, scratches, etc. A nonlinear phase distorts the proper registration of different frequency components that make up the lines and scratches. This distorts the signal shape in various ways, including blurring.

It is simple to design zero-phase FIR filters. It is impossible, however, for a single recursively computable IIR filter to have zero phase. To have zero phase, $h(n_1, n_2)$ must be equal to $h_d(-n_1, -n_2)$. An IIR filter requires an infinite extent $h(n_1, n_2)$, and output mask must have a wedge support. These requirements cannot all be satisfied at the same time. It is possible, however, to achieve zero-phase by using more than one IIR filter. A method particularly well suited to spatial domain design is to divide $h_d(n_1, n_2)$ into different regions, design an IIR filter to approximate $h_d(n_1, n_2)$ in each region, and then combine the filters by using a parallel structure.

Suppose we have a desired $h_d(n_1, n_2)$, we assume that

$$h_d(n_1, n_2) = h_d(-n_1, -n_2). \quad (13)$$

We can divide $h_d(n_1, n_2)$ into an even number of regions: Two, four, six, eight, or more.

Suppose we divide $h_d(n_1, n_2)$ into four regions by

$$h_d^I(n_1, n_2) = h_d(n_1, n_2) w(n_1, n_2), \quad (14a)$$

$$h_d^{II}(n_1, n_2) = h_d(n_1, n_2) w(-n_1, n_2), \quad (14b)$$

$$h_d^{III}(n_1, n_2) = h_d(n_1, n_2) w(-n_1, -n_2), \quad (14c)$$

and

$$h_d^{IV}(n_1, n_2) = h_d(n_1, n_2) w(n_1, -n_2), \quad (14d)$$

where $w(n_1, n_2)$ is a first-quadrant support sequence given by

$$w(n) = \begin{cases} 1, & n_1 \geq 1, n_2 \geq 1 \\ \frac{1}{2}, & n_1 \geq 1, n_2 = 0 \\ \frac{1}{2}, & n_1 = 0, n_2 \geq 1 \\ \frac{1}{4}, & n_1 = n_2 = 0 \end{cases} \quad (15)$$

Suppose we use one of the spatial IIR filter design techniques discussed earlier to design $H^I(z_1, z_2)$ that approximates $h_d^I(n_1, n_2)$. Similarly, suppose we have designed $H^{II}(z_1, z_2)$ that approximates $h_d^{II}(n_1, n_2)$. From (13) and (14),

$$h_d^{III}(n_1, n_2) = h_d^I(-n_1, -n_2). \tag{16}$$

Therefore, $H^{III}(z_1, z_2)$ that approximates $h_d^{III}(n_1, n_2)$ can be obtained from $H^I(z_1, z_2)$ by

$$H^{III}(z_1, z_2) = H^I(z_1^{-1}, z_2^{-1}). \tag{17}$$

Similarly, $H^{IV}(z_1, z_2)$ can be obtained from $H^{II}(z_1, z_2)$ by

$$H^{IV}(z_1, z_2) = H^{II}(z_1^{-1}, z_2^{-1}). \tag{18}$$

Since $H^I(z_1, z_2)$, $H^{II}(z_1, z_2)$, $H^{III}(z_1, z_2)$, and $H^{IV}(z_1, z_2)$ approximate $H^I(z_1, z_2)$, $h_d^{II}(n_1, n_2)$, $h_d^{III}(n_1, n_2)$ and $h_d^{IV}(n_1, n_2)$, respectively, $h_d(n_1, n_2)$ will be approximated by $H(z_1, z_2)$ given by

$$\begin{aligned} H(z_1, z_2) &= H^I(z_1, z_2) + H^{II}(z_1, z_2) \\ &\quad + H^{III}(z_1, z_2) + H^{IV}(z_1, z_2) \\ &= H^I(z_1, z_2) + H^{II}(z_1, z_2) \\ &\quad + H^I(z_1^{-1}, z_2^{-1}) + H^{II}(z_1^{-1}, z_2^{-1}). \end{aligned} \tag{19}$$

$H(z_1, z_2)$ has zero phase since $H(z_1, z_2) = H(z_1^{-1}, z_2^{-1})$. The system in (19) can be implemented by using a parallel structure as shown in **Figure 1**. The input is filtered by each of the four recursively computable systems, and the results are combined to produce the output.

If $h_d(n_1, n_2)$ has fourfold symmetry,

$$h_d^{II}(n_1, n_2) = h_d^I(-n_1, n_2), \tag{20}$$

then $H^{II}(z_1, z_2)$ can be determined from $H^I(z_1, z_2)$ by

$$H^{II}(z_1, z_2) = H^I(z_1^{-1}, z_2). \tag{21}$$

In this case, $H(z_1, z_2)$ in (19) is given by

$$\begin{aligned} H(z_1, z_2) &= H^I(z_1, z_2) + H^I(z_1^{-1}, z_2) \\ &\quad + H^I(z_1^{-1}, z_2^{-1}) + H^I(z_1, z_2^{-1}). \end{aligned} \tag{22}$$

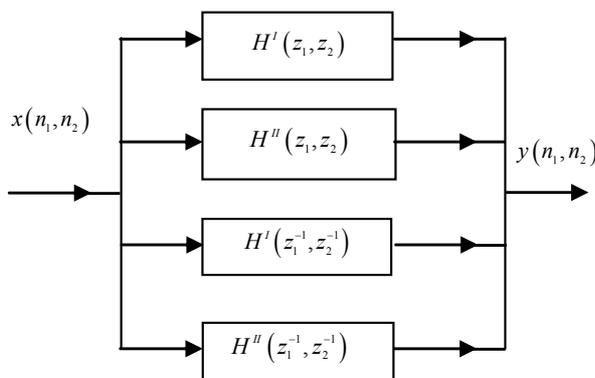


Figure 1. implementation of $H(z_1, z_2)$ using a parallel structure.

From (22), only one filter needs to be designed in this case.

2.4. Frequency Domain Design by the Iterative Semi-Definite Programming

Semi-definite programming (SDP) has recently attracted a great deal of research interest. Among other things, the optimization tool was proven to be applicable to the design of various types of FIR digital filters. An attempt on extending the SDP approach to 2-D IIR filters is made in [3]. Throughout this section the IIR filters are assumed to have separable denominators. This assumption simply imposes a constraint on the type of IIR filters being quadratically symmetric. Nevertheless, this class of filters is broad enough to cover practically all types of IIR filters that have been found useful in image/video [4].

Consider a quadratically symmetric 2-D IIR digital filter whose transfer function is given by

$$H(z_1, z_2) = \frac{B(z_1, z_2)}{A(z_1)A(z_2)}, \tag{23}$$

where

$$B(z_1, z_2) = \sum_{k_1=0}^n \sum_{k_2=0}^n b(k_1, k_2) z_1^{-k_1} z_2^{-k_2}$$

and $A(z) = \sum_{k=0}^r a(k)z^{-k}$, $a(0) = 1$.

Since the filter is quadratically symmetric, we have $b(k_1, k_2) = b(k_2, k_1)$. As a result, there are only $r + (n+1)(n+2)/2$ unknown variables in (23), which form a $[r + 0.5(n+1)(n+2)]$ -dimension vector

$$x = [a_1 \cdots a_r b_{00} \cdots b_{nn} b_{10} b_{20} b_{21} \cdots b_{n0} \cdots b_{n,n-1}]^T \tag{24}$$

where $a_i = a(i)$, $b_{ij} = b(i, j)$. Denote the vector x in the k^{th} iteration as x_k and the frequency response of the filter for $x = x_k$ as $H(e^{j\omega_1}, e^{j\omega_2}, x_k)$. In the neighborhood of x_k , the design variable can be expressed as $x = x_k + \delta$.

The transfer function can be approximated in terms of a linear function of δ by

$$H(e^{j\omega_1}, e^{j\omega_2}, x) \approx H(e^{j\omega_1}, e^{j\omega_2}, x_k) + g_k^T \delta \tag{25}$$

where g_k is the gradient of $\text{de } H(e^{j\omega_1}, e^{j\omega_2}, x)$ for $x = x_k$.

Problem formulation

The min-max design is obtained as a solution of the following optimization problem:

$$\text{Minimize } \hat{c}^T \hat{\delta} \tag{26}$$

Subject to:

$$\begin{bmatrix} S_k & 0 \\ 0 & Y_k \end{bmatrix} \succeq 0 \tag{27}$$

with

$$\hat{c} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad \hat{\delta} = \begin{bmatrix} \mu \\ \delta \end{bmatrix}, \quad (28)$$

where μ is treated as an additional design variable, and

$$S_k = \text{diag} \left\{ \Phi_k(\omega_1^{(1)}, \omega_2^{(1)}), \dots, \Phi_k(\omega_1^{(L)}, \omega_2^{(M)}) \right\}, \quad (29)$$

$$\Phi_k(\omega_1, \omega_2) = \begin{bmatrix} I & Q_k^{\frac{1}{2}} \delta \\ \delta^T Q_k^{\frac{1}{2}} & \mu - 2\delta^T q_k - c_k \end{bmatrix} \succeq 0, \quad (30)$$

$$Q_k = W(\omega_1, \omega_2) \Re e(\bar{g}_k g_k^T). \quad (31)$$

$W(\omega_1, \omega_2) \geq 0$ is a weighting function, $\Re e(\cdot)$ the real part of (\cdot) , and

$$q_k = W(\omega_1, \omega_2) \cdot \Re e \left\{ \left[H(e^{j\omega_1}, e^{j\omega_2}, x_k) - H_d(\omega_1, \omega_2) \right] \bar{g}_k \right\}, \quad (32)$$

$$c_k = e(\omega_1, \omega_2, x_k) \quad (33)$$

$H_d(\omega_1, \omega_2)$ is the desired frequency response, for $(\omega_1, \omega_2) \in \Omega$, where $\Omega = \{(\omega_1, \omega_2) : -\pi \leq \omega_1, \omega_2 \leq \pi\}$, and

$$Y_k = \begin{bmatrix} P^{-1} - \tau I_r & D_k \\ D_k^T & P - \tau I_r \end{bmatrix} \succeq 0, \quad (34)$$

where P is defined below and τ is a positive scalar that stability margin of the filter, and

$$D_k = \begin{bmatrix} -(a_k + \delta_1)^T \\ \hat{I}_r \end{bmatrix}. \quad (35)$$

Denote the vectors formed from the first r components of $x_k + \delta$ by $a_k + \delta_1$. Since the denominator of $H(z_1, z_2)$ is separable, it can be shown that the IIR filter with coefficient vector $x_k + \delta$ is stable if and only if the magnitudes of the eigenvalues of matrices D_k are strictly less than one, where \hat{I}_r denote a matrix of size $(r-1) \times r$ obtained by augmenting the identity matrix with a zero column on the right. Applying the well-known Lyapunov theory [5], one concludes that matrix D_k is stable if and only if there exists a positive definite matrix P such that

$$P - D_k^T P D_k \succ 0, \quad (36)$$

where $M \succ 0$ denotes that matrix M is positive definite. The matrix P in (34) is not considered as a design variable. Rather, this positive definite matrix is fixed in each iteration and can be obtained by solving the Lyapunov equation

$$P - \hat{D}_k^T P \hat{D}_k = I, \quad (37)$$

where

$$\hat{D}_k = \begin{bmatrix} -a_k^T \\ \hat{I}_r \end{bmatrix}. \quad (38)$$

with P fixed in Y_k , the minimization problem in (26) and (27) is an SDP problem of size $1 + r + 0.5(n+1)(n+2)$.

Design steps

Given the order of the IIR filter (n, r) and the desired frequency response $H_d(\omega_1, \omega_2)$.

Step 1: The proposed design method starts with an initial point x_0 that corresponds to a stable filter obtained using a conventional method.

Step 2: With this x_0 , a positive definite matrix P can be obtained by solving the Lyapunov equation (37), and the quantities Q_k , q_k , and c_k can be evaluated by using (31)-(33).

Step 3: Next we solve the SDP problem in (26) and (27).

Step 4: The solution $\hat{x}^* = [\mu^* \ \delta^{*T}]^T$ obtained can be used to update x_0 to $x_1 = x_0 + \delta^*$. The iteration continues until $\|\delta^*\|$ is less than a prescribed tolerance ε .

3. Order Reduction

It is often desirable to represent a high order system by a lower order system. A suitable model reduction procedure should provide a model that approximates the original well, it should produce stable models from a stable original, and it should be able to be implemented on a computer with high computational efficiency and reduced memory requirements. The reduction of models in the state space (SS) realization environment has definite advantages. It is possible to apply the vast knowledge of matrix theory in the analysis, while the non-uniqueness of SS realization allows to choose of using one that is better suited for the purpose at hand [6,7].

3.1. State-Space Model

Roesser's model is the following [6,7]:

$$x(i, j) = \begin{bmatrix} x^h x(i, j) \\ x^v x(i, j) \end{bmatrix}, \quad (39a)$$

where x is the local state, x^h , an n -vector, is the horizontal state, x^v , an m -vector, is the vertical state, and

$$\begin{bmatrix} x^h(i+1, j) \\ x^v(i, j+1) \end{bmatrix} = \begin{bmatrix} A_1 & A_2 \\ A_3 & A_4 \end{bmatrix} \begin{bmatrix} x^h(i, j) \\ x^v(i, j) \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u(i, j), \quad (39b)$$

$$y(i, j) = [C_1 \ C_2] \begin{bmatrix} x^h(i, j) \\ x^v(i, j) \end{bmatrix} + du(i, j), \quad (39c)$$

$$(i, j) \geq (0, 0),$$

where u , the input, is an l -vector and y , the output, is a p -vector. Clearly x^h , the horizontal state, is propagated horizontally, and x^v , the vertical state is propagated vertically by first-order difference equations.

The 2-D transfer function can be written as:

$$H(z_1, z_2) = C \left[\begin{pmatrix} z_1 I & 0 \\ 0 & z_2 I \end{pmatrix} - A \right]^{-1} B + d. \quad (40)$$

It is clear that there is a one-to-one correspondence between Roesser's model and circuit implementations with delay elements z_1^{-1} and z_2^{-1} .

3.2. Minimal State Space Realization

The minimal state-space realizations are not possible for all 2-D transfer functions [8]. However, minimal state-space realizations have been determined for a system with separable denominator [9].

Consider the linear time-invariant 2-D system, described by the spatial transfer function [10]

$$H(z_1^{-1}, z_2^{-1}) = \frac{\sum_{k_1=0}^n \sum_{k_2=0}^m b(k_1, k_2) z_1^{-k_1} z_2^{-k_2}}{\sum_{k_1=0}^n \sum_{k_2=0}^m a(k_1, k_2) z_1^{-k_1} z_2^{-k_2}}. \quad (41)$$

The numerator coefficients of (41) can be arbitrary, while the denominator coefficients satisfy the following relationship:

$$a(k_1, k_2) = a(k_1, 0)a(0, k_2), \quad (42)$$

with $a(0, 0) = 1$.

The state-space model sought is of the Givone-Roesser type [10], described with $l = p = 1$, and $A_1, A_2, A_3, A_4, B_1, B_2, C_1$, and C_2 of dimension $(n \times n)$, $(n \times m)$, $(m \times n)$, $(m \times m)$, $(n \times 1)$, $(m \times 1)$, $(n \times 1)$, and $(m \times 1)$, respectively.

The minimal realization in this case requires only $n + m$ dynamic elements. A realization of the state-space can be written as:

$$A_1 = \begin{bmatrix} -a(0,1) & \cdots & \cdots & \cdots & \cdots & -a(n-1,1) \\ 1 & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & \ddots & \ddots & & & \vdots \\ \vdots & \ddots & \ddots & \ddots & & \vdots \\ 0 & \cdots & \cdots & 0 & 1 & 0 \end{bmatrix},$$

$$A_2 = \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & & \vdots \\ 0 & \cdots & 0 \end{bmatrix},$$

$$A_3 = \begin{bmatrix} h(1,1) & \cdots & h(1,n) \\ \vdots & \ddots & \vdots \\ h(m,1) & \cdots & h(m,n) \end{bmatrix},$$

where

$$\begin{aligned} h(1,1) &= b(1,1) + a(1,0)a(0,1)b(0,0) \\ &\quad - (b(1,0)a(0,1) + a(1,0)b(0,1)), \\ &\quad \vdots \\ h(m,n) &= b(m,n) + a(m,0)a(0,n)b(0,0) \\ &\quad - (b(m,0)a(0,n) + a(m,0)b(0,n)), \end{aligned}$$

and

$$A_4 = \begin{bmatrix} -a(1,0) & 1 & 0 & \cdots & 0 \\ \vdots & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \vdots & \vdots & & \ddots & 0 \\ \vdots & \vdots & & \ddots & 1 \\ -a(m,0) & 0 & \cdots & \cdots & 0 \end{bmatrix},$$

$$B_1 = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix},$$

$$B_2 = \begin{bmatrix} b(1,0) - a(1,0)b(0,0) \\ \vdots \\ b(m,0) - a(m,0)b(0,0) \end{bmatrix},$$

$$C_1 = [b(0,1) - a(0,1)b(0,0) \ \cdots \ b(0,n) - a(0,n)b(0,0)],$$

$$C_2 = [1 \ 0 \ \cdots \ 0], \text{ and } d = b(0,0) \text{ [11,12].}$$

3.3. Model Order Reduction Methods

The most popular 1-D model reduction techniques are based on the concept of balanced realization which was originally proposed by Moore [13]. Given a discrete system, its balanced realization describes the system in a state-space representation in which the importance of the i^{th} state variable can be measured by the i^{th} Hankel singular value of the system. This suggest that one way of obtaining a low-order approximation of a state-space model is to form a balanced realization and then to retain those states corresponding to r largest Hankel singular values, where r is the order of the reduced-order system.

One of the problems in the study of 2-D model reduction is to extend the balanced realization concept to the 2-D case. Since a balanced realization is essentially determined by the controllability and observability Gramians of the system, and since there are several types of Gramians of the system that can be properly defined for a given 2-D system, there are different types of balanced realizations for a 2-D discrete system, leading to different balanced approximations [14].

3.3.1. Three Types of Gramians for 2-D Discrete Systems

Consider the Givone-Roesser state-space model of a SISO system described in (39), and define [14]

$$F(z_1, z_2) = (I(z_1, z_2) - A)^{-1} B, \quad (43a)$$

$$G(z_1, z_2) = C(I(z_1, z_2) - A)^{-1}, \quad (43b)$$

where $I(z_1, z_2) = z_1 I_n \oplus z_2 I_m$.

The first type of 2-D Gramians, known as the pseudo-controllability and observability Gramians [15], are defined as

$$P^p = \frac{1}{(2\pi j)^2} \oint_{|z_1|=1} \oint_{|z_2|=1} F(z_1, z_2) F^*(z_1, z_2) \frac{dz_2}{z_2} \frac{dz_1}{z_1},$$

$$Q^p = \frac{1}{(2\pi j)^2} \oint_{|z_1|=1} \oint_{|z_2|=1} G^*(z_1, z_2) G(z_1, z_2) \frac{dz_2}{z_2} \frac{dz_1}{z_1}.$$

The second type of Gramians, known as the structured controllability and observability Gramians [16], are defined by the positive definite solutions set of P^s and Q^s Lyapunov inequalities

$$AP^s A^T - P^s + BB^T < 0,$$

$$A^T Q^s A - Q^s + C^T C < 0.$$

Note that the positive definite matrices P^s and Q^s , if they exist, are not unique. This leads to the nonuniqueness of structurally balanced realizations that are based on P^s and Q^s .

The third type of Gramians, known as quasi-controllability and observability Gramians [17], are defined by the positive definite block-diagonal matrices $P^q = \text{diag}\{P_1, P_2\}$ and $Q^q = \text{diag}\{Q_1, Q_2\}$ where P_i and Q_i ($i = 1, 2$) satisfy the Lyapunov equations:

$$A_1 P_1 A_1^T - P_1 + B_1 B_1^T + A_2 P_2 A_2^T = 0,$$

$$A_4 P_2 A_4^T - P_2 + B_2 B_2^T + A_3 P_1 A_3^T = 0,$$

$$A_1^T Q_1 A_1 - Q_1 + C_1^T C_1 + A_3^T Q_2 A_3 = 0,$$

$$A_4^T Q_2 A_4 - Q_2 + C_2^T C_2 + A_2^T Q_1 A_2 = 0.$$

In terms of computation complexity, the structured Gramians are the most expensive to evaluate while the

quasi-Gramians are the most economical [14].

3.3.2. Balanced Realizations Approximation

Since there are at least three types of Gramians, one can accordingly derive three different types of balanced realizations. In effect, once a certain type of Gramians is chosen, the upper left and lower right diagonal blocks of the Gramians are used to compute the transformation matrix $T = T_1 \oplus T_2$, by using, for example, Laub's algorithm [18] such as the realization characterized by $\{T^{-1}AT, T^{-1}B, CD, d\}$ is balanced.

A reduced-order system of order (r_1, r_2) , denoted by $\{A_r, B_r, C_r, d\}$ can be obtained by truncating the matrices A , B , and C as

$$A_r = \begin{bmatrix} A_{1r} & A_{2r} \\ A_{3r} & A_{4r} \end{bmatrix}, B_r = \begin{bmatrix} B_{1r} \\ B_{2r} \end{bmatrix}, C_r = [C_{1r} \quad C_{2r}],$$

where [14]

$$A_{1r} = A_1(1:r_1, 1:r_1), A_{2r} = A_2(1:r_1, 1:r_2),$$

$$A_{3r} = A_3(1:r_2, 1:r_1), A_{4r} = A_4(1:r_2, 1:r_2),$$

$$B_{1r} = B_1(1:r_1), B_{2r} = B_2(1:r_2),$$

$$C_{1r} = C_1(1:r_1), C_{2r} = C_2(1:r_2).$$

3.3.3. Iterative Algorithm for Computing the Quasi-Gramians

An iterative method for the computation of Quasi-Gramians is described, where each iteration involves solving two 1-D Lyapunov equations. For a 2-D stable system, the algorithm converges very quickly to the 2-D quasi-Gramians [19].

Step 1: Set $P_2^{(0)} = Q_2^{(0)} = 0$, and $k = 1$.

Step 2: Solve the following 1-D Lyapunov equations for $P_1^{(k)}$ and $Q_1^{(k)}$:

$$A_1 P_1^{(k)} A_1^T - P_1^{(k)} + F_1 = 0, \quad (44a)$$

$$A_1 Q_1^{(k)} A_1^T - Q_1^{(k)} + G_1 = 0, \quad (44b)$$

where

$$F_1 = B_1 B_1^T + A_2 P_2^{(k-1)} A_2^T,$$

$$G_1 = C_1^T C_1 + A_3^T Q_2^{(k-1)} A_3.$$

Step 3: Solve the following 1-D Lyapunov equations for $P_2^{(k)}$ and $Q_2^{(k)}$:

$$A_4 P_2^{(k)} A_4^T - P_2^{(k)} + F_2 = 0, \quad (44c)$$

$$A_4 Q_2^{(k)} A_4^T - Q_2^{(k)} + G_2 = 0, \quad (44d)$$

where

$$F_2 = B_2 B_2^T + A_3 P_1^{(k)} A_2^T,$$

$$G_2 = C_2^T C_2 + A_2^T Q_1^{(k)} A_2.$$

Step 4: Set $k = k + 1$ and repeat Steps 2 and 3 until

$$\|P_i^{(k)} - P_i^{(k-1)}\| < \varepsilon, \quad (i = 1, 2),$$

$$\|Q_i^{(k)} - Q_i^{(k-1)}\| < \varepsilon, \quad (i = 1, 2).$$

where ε is a prescribed tolerance [19].

4. Illustrative Simulations

We have divided the simulation into two parts: 2D-filter

design and order reduction. The interpretation of the results is given at the end of this section.

Part 1: 2D-filter design

The design has been performed in two domains (see **Figures 2-7**):

- Spatial domain by two methods: Prony’s method and the Iterative method.

The numerator $b(n_1, n_2)$, and the denominator $a(n_1, n_2)$, matrices are generated, then we use the func-

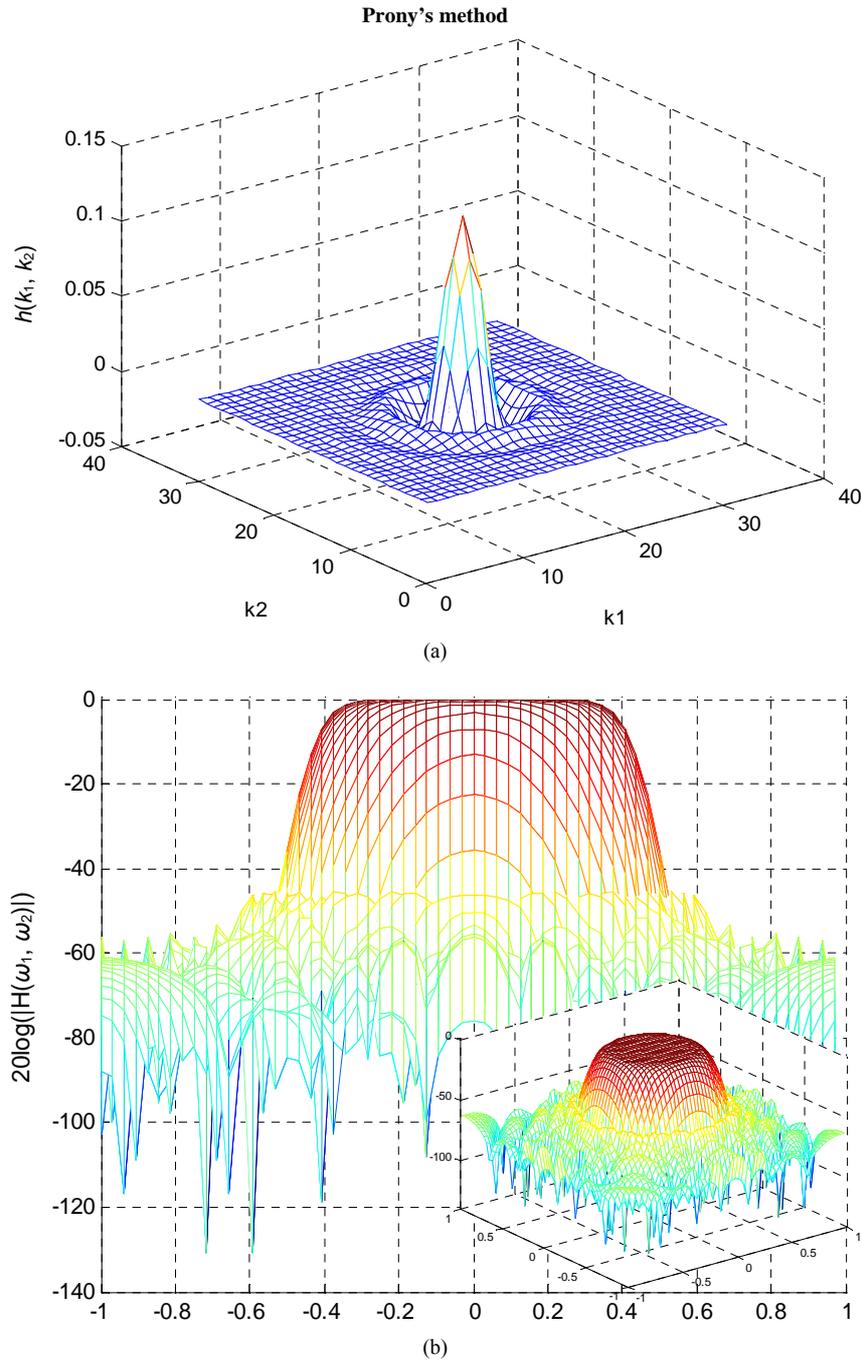


Figure 2. Zero-phase lowpass IIR filter. (a) Impulse response; (b) Amplitude responses (in decibels), “X-Z” perspective.

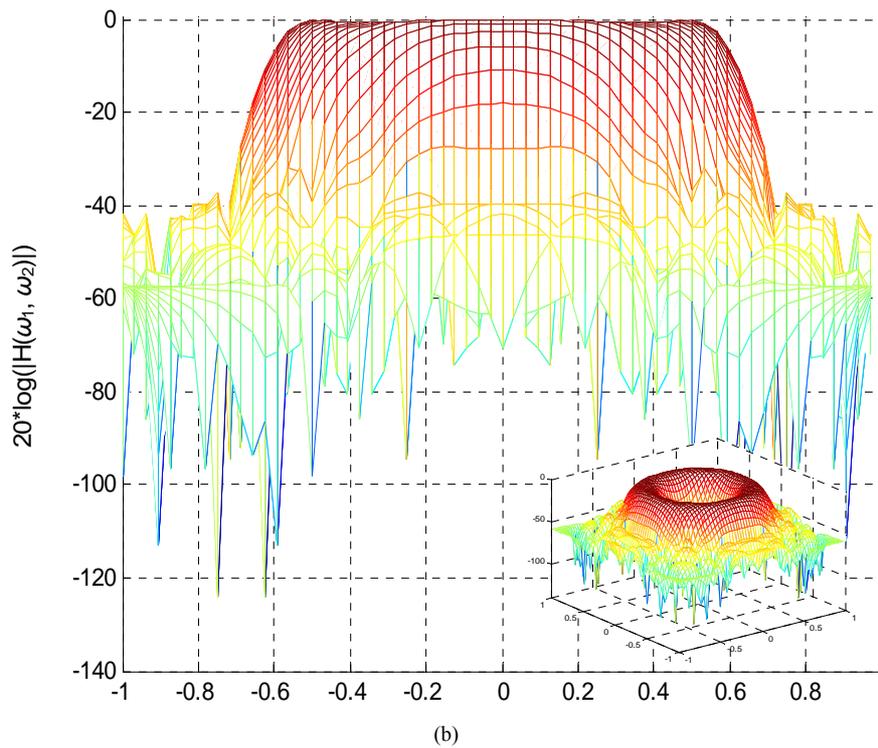
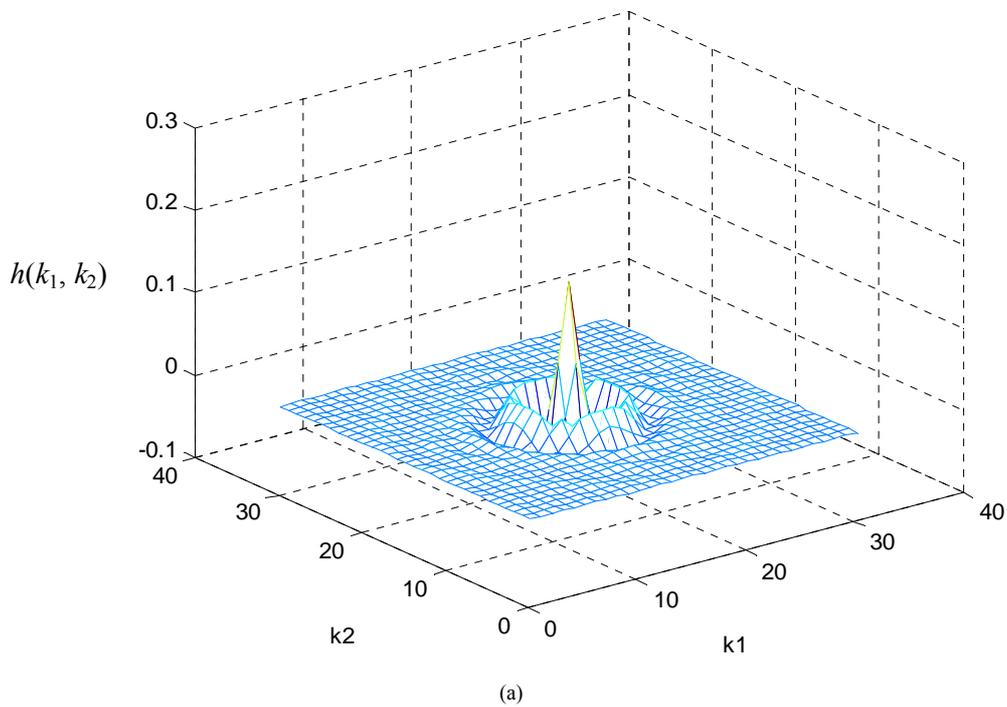


Figure 3. Zero-phase bandpass IIR filter. (a) Impulse response; (b) Amplitude responses (in decibels), “X-Z” perspective.

tion `Impulse_2D.m` to produce the impulse response and frequency response.

- Frequency domain, we use the method of Semidefinite Programming (SDP) to do the same.

For the Prony’s method and Iterative method the IIR

digital filters are specified by the following templates:

Filter’s order $n = 5$, $m = 5$, (the a and b matrix dimension).

The pass band and stop band corresponding to each type of filter are: Lowpass: $Rp = [0 \ 0.4]$, and $Rs = [0.5 \ 1]$.

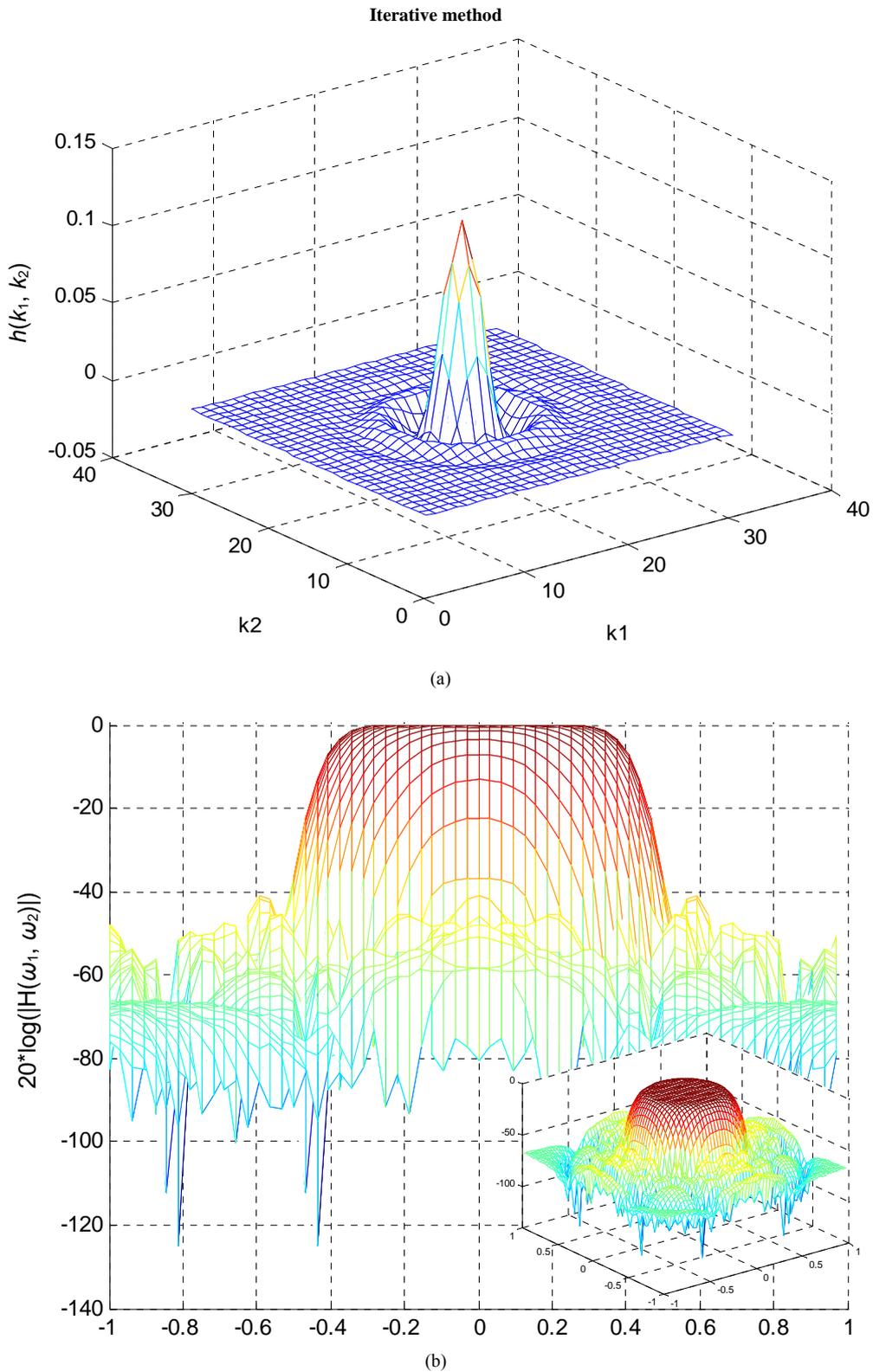
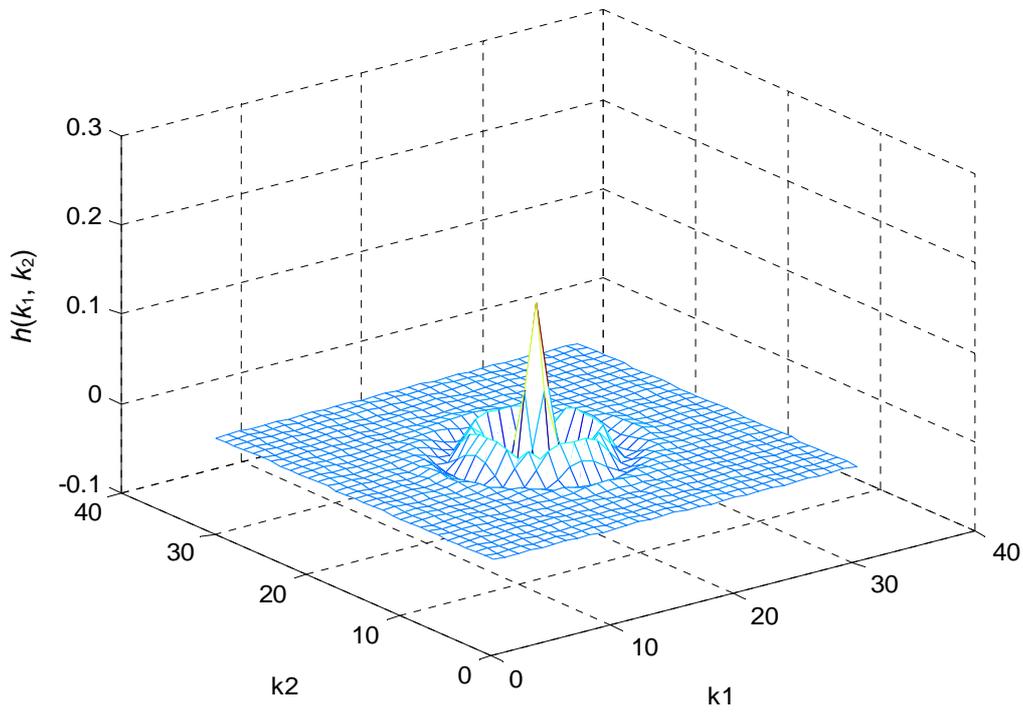


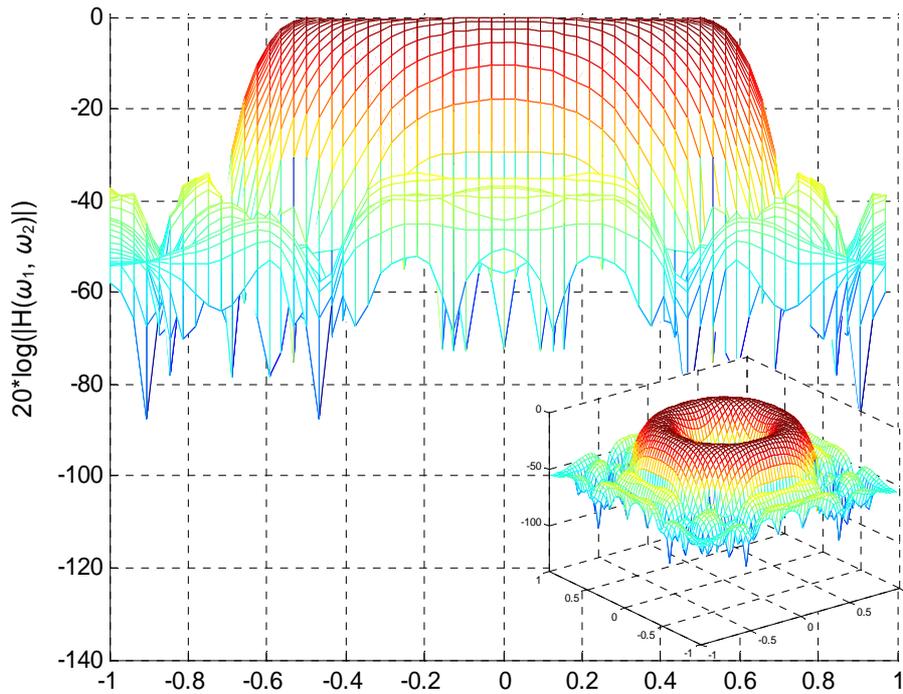
Figure 4. Zero-phase lowpass IIR filter. (a) Impulse response; (b) Amplitude responses (in decibels), “X-Z” perspective.

Bandpass: $R_p = [0.4 \ 0.6]$, $R_{s1} = [0 \ 0.3]$ and $R_{s2} = [0.7 \ 1]$.
 For the SDP method, let IIR digital filters be specified

by the following templates:
 Filter order $n = 13$, $m = 13$ (the a and b matrix



(a)



(b)

Figure 5. Zero-phase bandpass IIR filter. (a) Impulse response; (b) Amplitude responses (in decibels), “X-Z” perspective.

dimension).

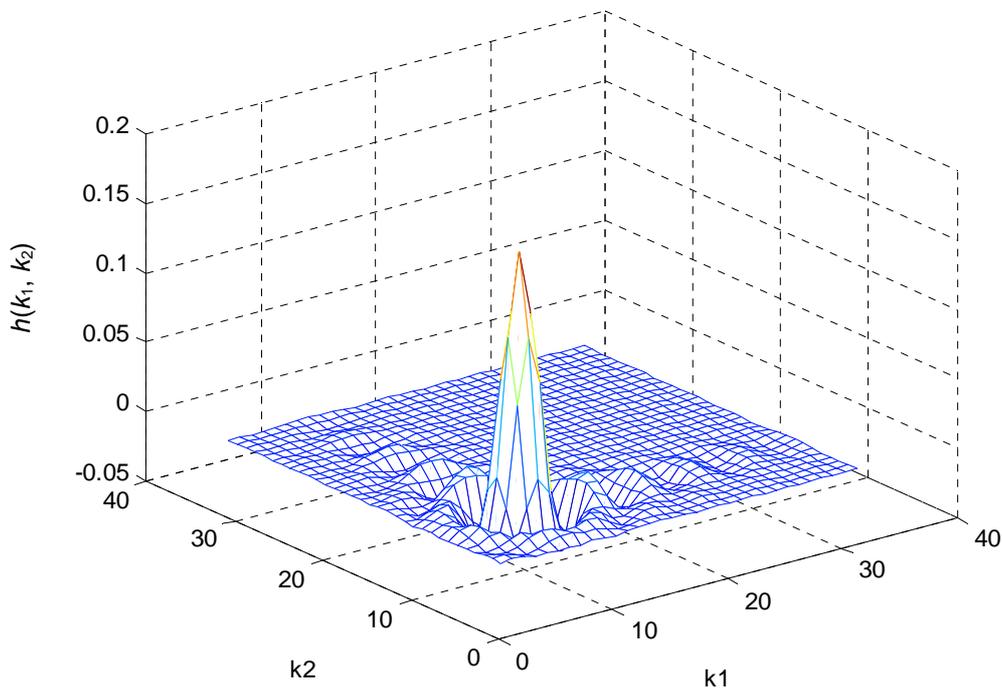
The passband and stopband corresponding to each type of filter are:

Lowpass: $R_p = [0 \ 0.4]$, and $R_s = [0.5 \ 1]$.

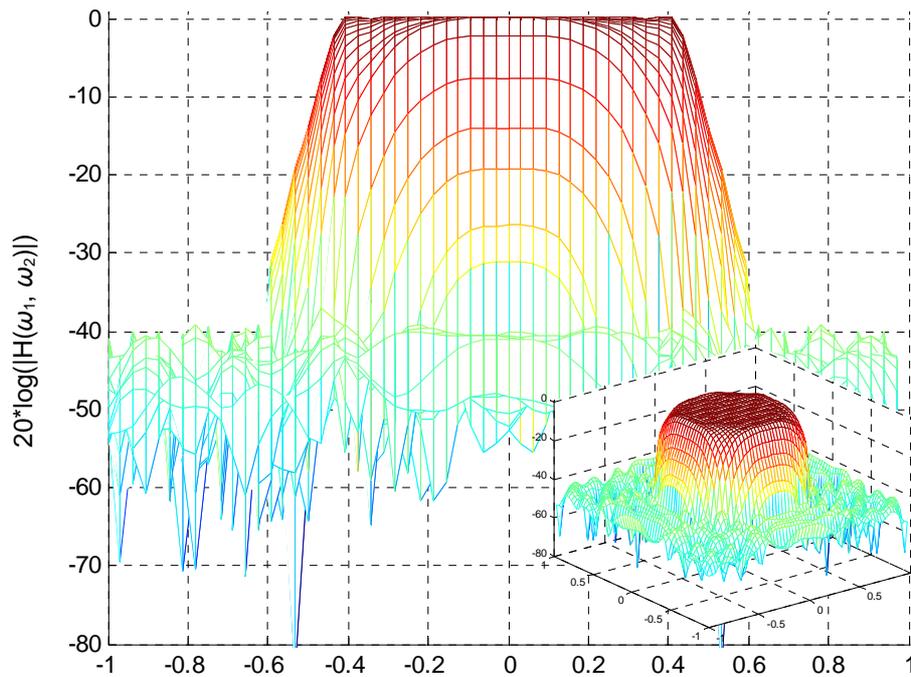
Bandpass: $R_p = [0.48 \ 0.55]$, $R_{s1} = [0 \ 0.3]$ and $R_{s2} = [0.7 \ 1]$.

Number of iterations: $it = 2$.

Part 2: Order reduction



(a)



(b)

Figure 6. Lowpass IIR filter. (a) Impulse response; (b) Amplitude responses (in decibels), “X-Z” perspective.

We apply the method of Quasi-Gramians to the lowpass filter designed in the first part. First, we use our function `tf2ss_2D.m` to transform the transfer function matrices a and b to the state-space model (A, B, C, d) ; then we applied the Quasi-Gramians method to

produce a reduced-order model $\{A_r, B_r, C_r, d\}$, (see **Figures 8-11**).

Iterative method

The order of original filter is: $n = 5, m = 5$; the total number of coefficient is:

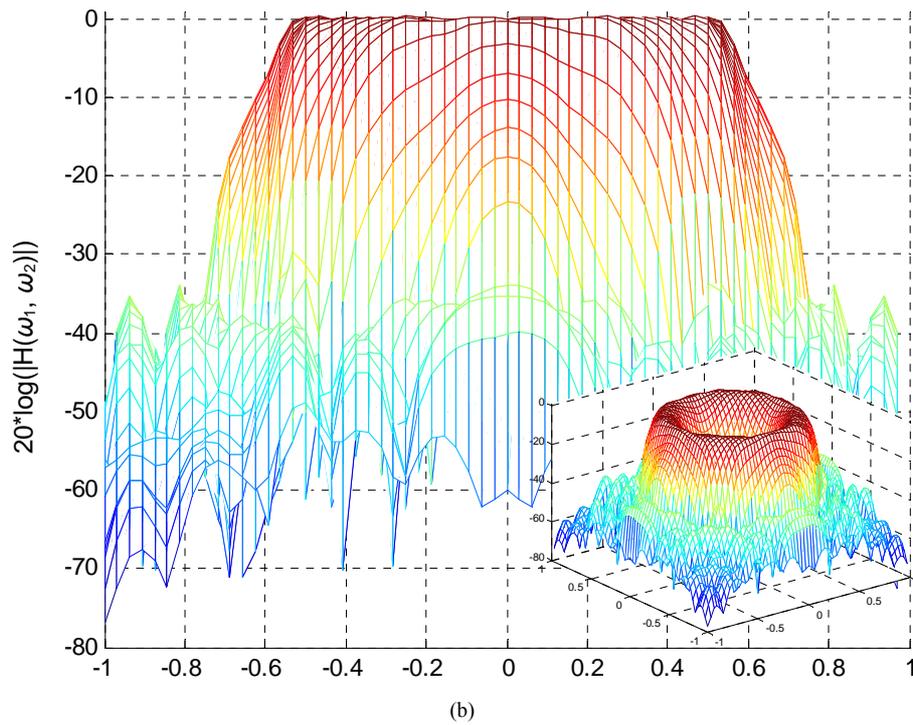
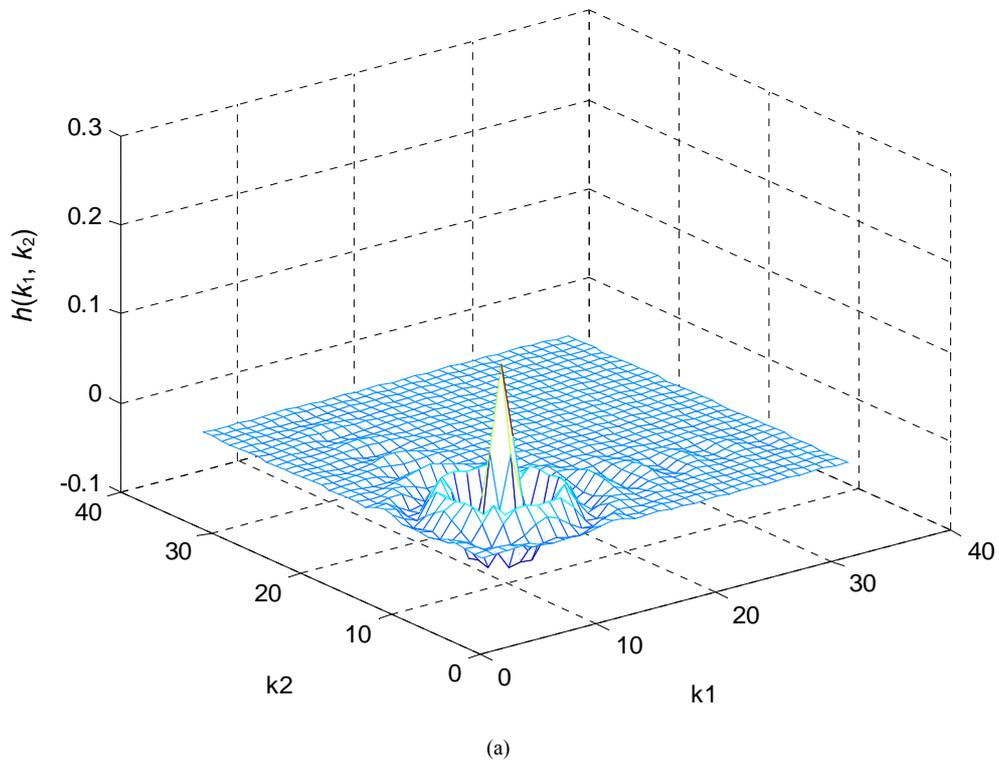


Figure 7. Bandpass IIR filter. (a) Impulse response; (b) Amplitude responses (in decibels), “X-Z” perspective.

$$((5 \times 5(\text{matrix}b) + 5 \times 5(\text{matrix}a))4 = 200).$$

The dimensions of the matrices A , B , and C are $(n + 2m) \times (n + 2m)$, $(n + 2m) \times 1$, and $1 \times (n + 2m)$,

respectively.

Number of iterations is: $it = 10$.

Order of reduced filter is: $r_1 = 4, r_2 = 5$.

- The total number of coefficient is

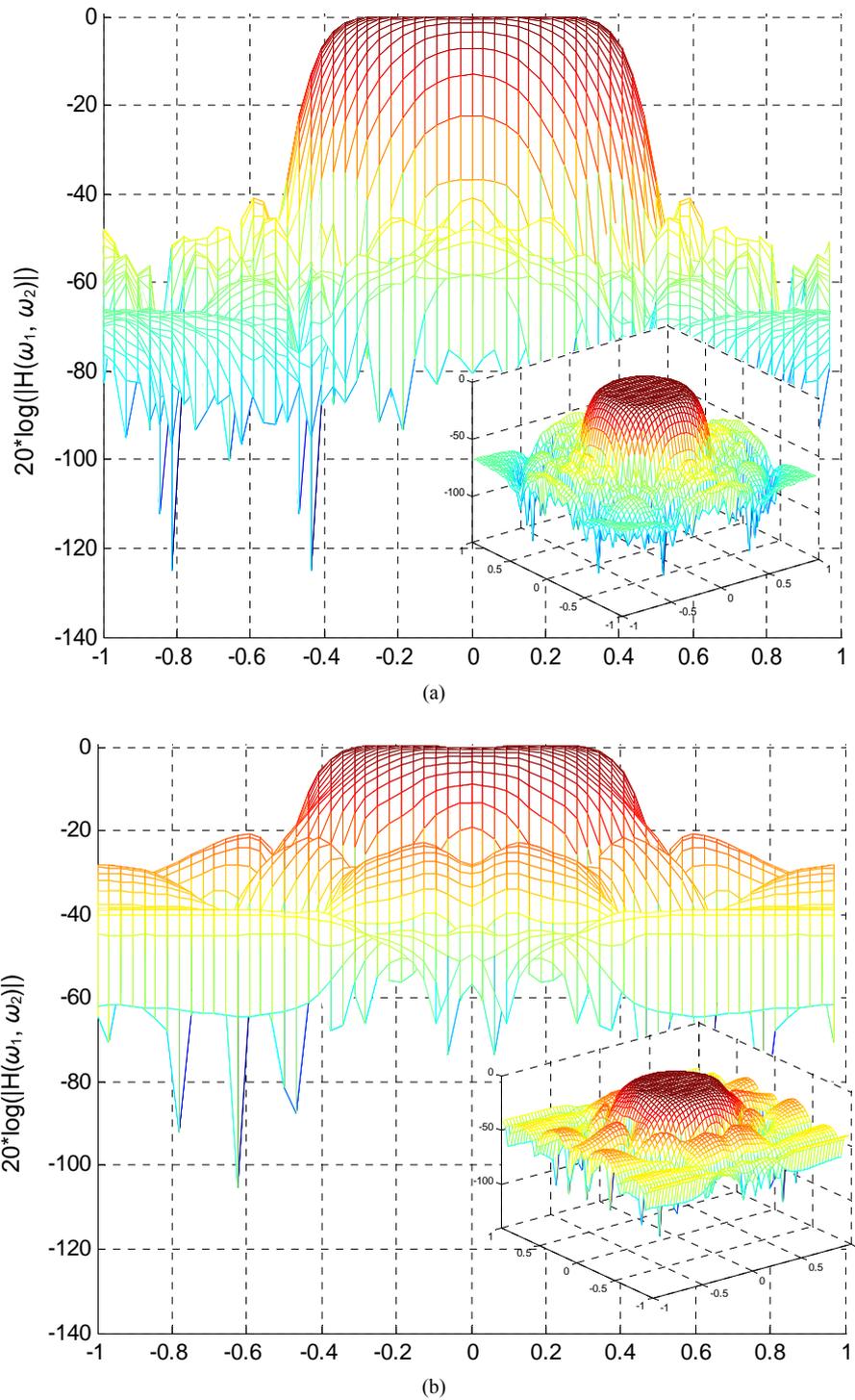


Figure 8. Lowpass IIR filter amplitude responses (in decibels), “X-Z” perspective. (a) Original filter 5×5 (200 coefficients); (b) Reduced-order filter 4×5 , (160 coefficients).

- $(4 \times 5(\text{matrix}b) + 4 \times 5(\text{matrix}a))4 = 160$.
 - Reduced-order filter 4×5 (160 coefficients).
- SDP method**
- The order of original filter is: $n = 13, m = 13$.
 - The total number of coefficient is:

- $(13 \times 13(\text{matrix}b) + 13 \times 13(\text{matrix}a)) = 338$.
- The dimensions of the matrices $A, B,$ and C are $(n + m) \times (n + m), (n + m) \times 1,$ and $1 \times (n + m),$ respectively (minimal realization).

Number of iterations is: $it = 2$.

Order of reduced filter is: $r_1 = 8, r_2 = 8$.

- The total number of coefficient is:
 $(8 \times 8(\text{matrix}b) + 8 \times 8(\text{matrix}a)) = 128$.

Interpretation

The results obtained lead to the following:

For the design, there is not much difference between the Prony method and the Iterative method. However, we can notice that there is a small improvement at stop-band (attenuation) when Iterative method is applied:

Prony: $\min(H(\omega_1, \omega_2)) = -49.09$ dB (lowpass filter);

Iterative: $\min(H(\omega_1, \omega_2)) = -55.77$ dB (lowpass filter).

And the results obtained with the SDP method are comparable to the other two methods, but we found a performance decrease in both bands (pass band and stop-band). It is possible to improve the results by increasing number of coefficients (n, r) or the number of iterations. A good aspect of this method is the stability, which can be measured by the maximum of the roots of the column or row vectors of the denominator matrix “ a ”: we found the following results:

$\max(\text{abs}(\text{roots}(a(:, 1)))) = 0.8926$ for the low-pass filter, and $= 0.8578$ for the bandpass filter.

where \max , abs , and roots are Matlab functions [20].

*The order reduction:

- Since the filter designed by Prony and Iterative

methods has a non-minimal realization, and the reduced filter can be unstable, the reduction results are not good (see **Figures 8 and 9**).

- But the stability is preserved in the SDP method:
 For the original filter, $\max(\text{abs}(\text{roots}(a(:, 1)))) = 0.8926$, and $= 0.9119$, for the reduced low-pass filter.

The results obtained show that the reduced low-pass filter is acceptable as the number of coefficients decreases from 338 to 128, and the max error between the reduced and original filter is $\max(E) \leq 0.02$ (cf. **Figure 11**).

5. Conclusions

In the design stage, stability and zero-phase of a 2-D IIR filter may become big problems. In 1-D we represent a pole or a zero by a point in a 2-D plan (real part and imaginary part), but in 2-D, the poles are surfaces in a 4-D plan, and so it is extremely difficult to test the stability and to stabilize an unstable filter. In image processing, the output image from a system (a filter in this case) can be deformed by the phase nonlinearity.

The reduction also suffers from some problems. The reduction in the state-space by conventional methods, for example, using a balanced transformation, requires a minimal realization of a transfer function in state-space, and then to find a method to compute the transformation matrix to a balanced model, which requires to maintain the stability and reduce as much as possible the computation effort.

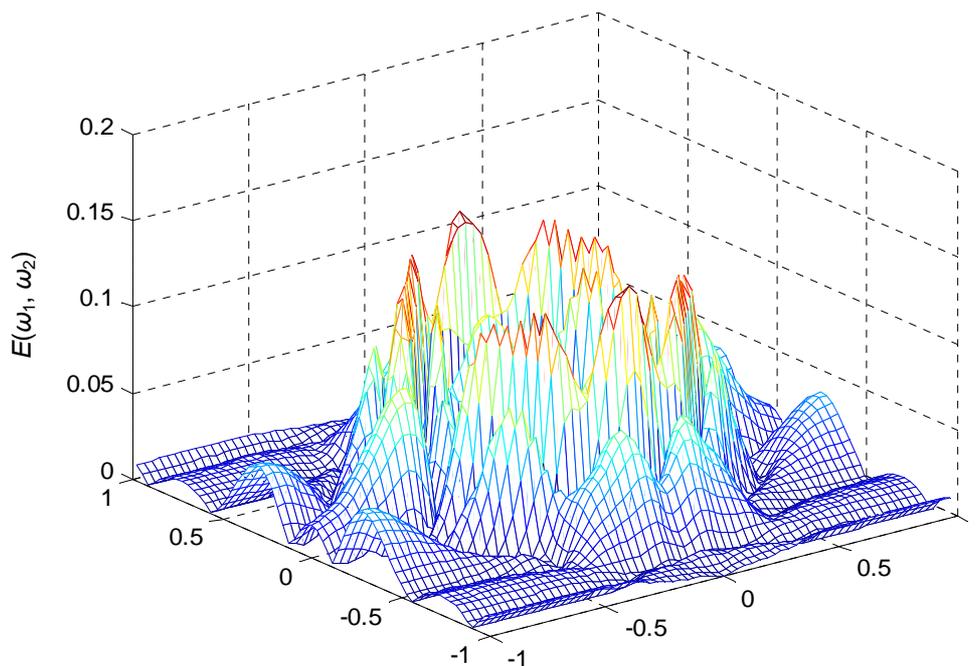


Figure 9. Error specter between original and reduced filter.

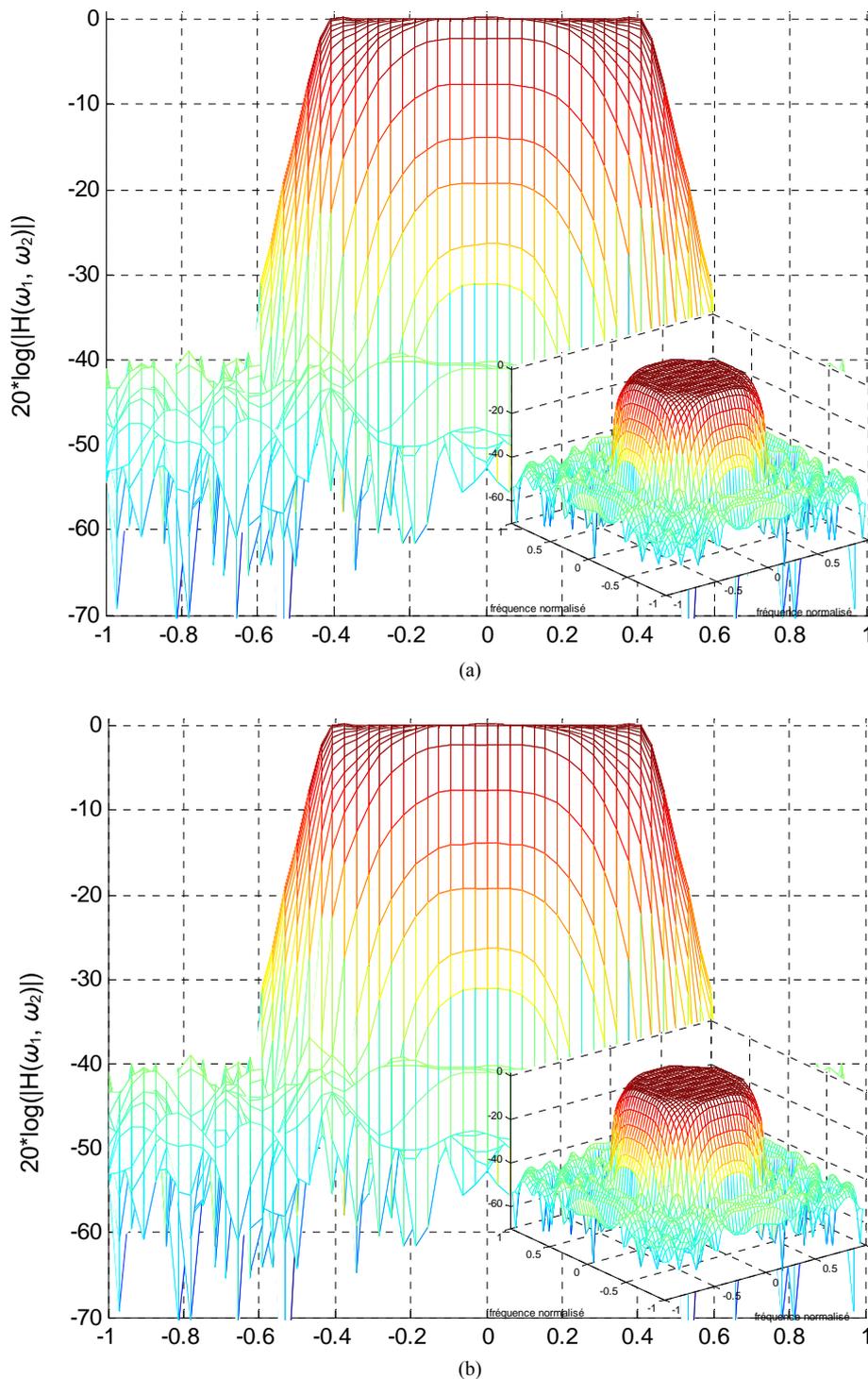


Figure 10. Lowpass IIR filter amplitude responses (in decibels), “X-Z” perspective. (a) Original filter 13×13 (338 coefficients); (b) Reduced-order filter 8×8 (128 coefficients).

- In this paper, we attempt to find a design method that solves these problems with conventional methods, such as Prony’s method or Iterative method, which are based on identification ideas. For a zero-phase filter, we divided the support region of the desired im-

pulse response on four regions, then we have synthesized the four filters (one filter in the case where $h_d(n_1, n_2)$ has a four-fold symmetry) by the Prony method or Iterative method. The result is a linear phase filter which has minimal order (the total num-

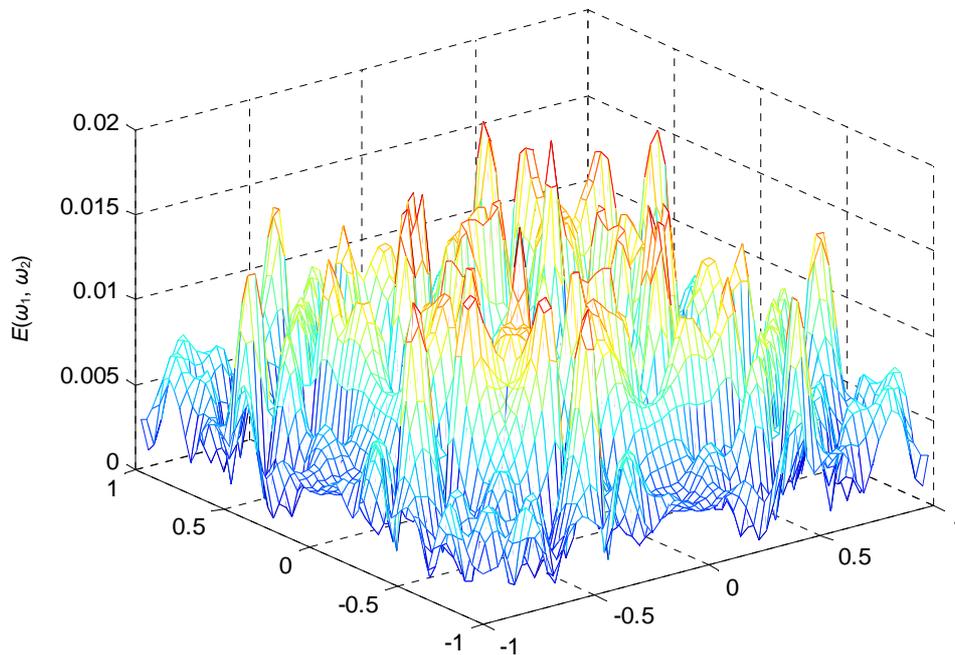


Figure 11. Error specter between original and reduced filter.

ber of coefficients is 200). In the method for Semidefinite Programming (the total number of coefficients is 338), but in this method the stability is guaranteed, with approximately linear phase.

- In the order reduction, we have chosen the method of Quasi-Gramians due to its simplicity. But this method has a disadvantage that it does not preserve the stability in general, but if the system has a separable denominator, the reduced system remains stable, as a filter designed by the SDP method.

REFERENCES

- [1] J. S. Lim, "Two-Dimensional Signal and Image Processing," Prentice Hall, Upper Saddle River, 1990.
- [2] R. C. Gonzalez and R. E. Woods, "Digital Image Processing," 3rd Edition, Prentice Hall, Upper Saddle River, 2007.
- [3] W. S. Lu, "Design of Stable 2-D IIR Digital Filters Using Iterative Semidefinite Programming," *Proceedings of ISCAS'00*, Geneva, 28-31 May 2000, pp. 395-398.
- [4] W. S. Lu, "A Unified Approach for the Design of 2-D Digital Filters via Semidefinite Programming," *IEEE Transactions on Circuits and Systems*, Vol. 49, June 2002, pp. 814-826. [doi:10.1109/TCSI.2002.1010036](https://doi.org/10.1109/TCSI.2002.1010036)
- [5] T. Kailath, "Linear Systems," Prentice Hall, Upper Saddle River, 1981.
- [6] K. Premaratne, E. I. Jury and M. Mansour, "An Algorithm for Model Reduction of 2-D Discrete Time Systems," *IEEE Transactions on Circuits and Systems*, Vol. 37, No. 9, 1990, pp. 1116-1132. [doi:10.1109/31.57600](https://doi.org/10.1109/31.57600)
- [7] S. Y. Kung, B. C. Levi, M. Morf and T. Kailath, "New Results in 2-D Systems Theory, Part II: 2-D State-Space Models-Realization and the Notions of Controllability, Observability, and Minimality," *Proceedings of IEEE*, Vol. 65, No. 6, 1977, pp. 945-961.
- [8] E. D. Sontag, "On First Order Equations for Multidimensional Filters," *IEEE Transactions on Acoustics, Speech and Signal Processing*, Vol. 26, No. 5, 1978, pp. 480-482.
- [9] S. Attasi, "Modélisation et traitement des suites à deux indices," Technical Report 31, INRIA, Le Chesnay, 1975.
- [10] J. L. Shanks, S. Treitel and J. H. Justice, "Stability and Synthesis of Tow-Dimensional Recursive Filters," *IEEE Transactions on Audio and Electroacoustics*, Vol. 20, No. 2, 1972, pp. 115-128.
- [11] D. Givone and R. P. Roesser, "Minimization of Multidimensional Linear Iterative Circuits," *IEEE Transactions on Computers*, Vol. 22, No. 7, 1973, pp. 673-678. [doi:10.1109/TC.1973.5009134](https://doi.org/10.1109/TC.1973.5009134)
- [12] G. E. Antoniou, P. N. Paraskevopoulos and S. J. Varoufakis, "Minimal State-Space Realization of Factorable 2-D Transfer Function," *IEEE Transactions on Circuits and Systems*, Vol. 35, No. 8, 1988, pp. 1055-1058. [doi:10.1109/31.1857](https://doi.org/10.1109/31.1857)
- [13] B. C. Moore, "Principal Component Analysis in Linear Systems: Controllability, Observability and Model Reduction," *IEEE Transactions on Automatic Control*, Vol. 26, No. 1, 1981, pp. 17-32. [doi:10.1109/TAC.1981.1102568](https://doi.org/10.1109/TAC.1981.1102568)
- [14] W. S. Lu, H. Luo and A. Antonio, "Recent Results on Model Reduction Methods for 2-D Discrete Systems," *IEEE Symposium on Circuits and Systems*, Atlanta, 12-15 May 1996, Vol. 2, pp. 348-351.
- [15] W.-S. Lu, E. B. Lee and Q. T. Zhang, "Balanced Approximation of Two-Dimensional and Delay Differential

- Systems,” *International Journal of Control*, Vol. 46, No. 6, 1987, pp. 2199-2218.
[doi:10.1080/00207178708934043](https://doi.org/10.1080/00207178708934043)
- [16] W. Wang, K. Glover, J. C. Doyle and C. Beck, “Model Reduction of LFT Systems,” *Proceedings of the 30th IEEE Conference on Decision and Control*, Brighton, 11-13 December 1991, pp. 1233-1238.
[doi:10.1109/CDC.1991.261574](https://doi.org/10.1109/CDC.1991.261574)
- [17] K. Zhou, J. L. Aravena, G. Gu and D. Xiong, “Two-Dimensional System Model Reduction by Quasi-Balanced Truncation and Singular Perturbation,” *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, Vol. 41, No. 9, 1994, pp. 593-602.
[doi:10.1109/82.326586](https://doi.org/10.1109/82.326586)
- [18] A. J. Laub, “On Computing Balancing Transformations,” *Proceedings of JACC*, San Francisco, Vol. 1, Paper FA8-E, 1980.
- [19] H. Luo, W. S. Lu and A. Antoniou, “Numerical Solution of the Two-Dimensional Lyapunov Equations and Application in Order Reduction of Recursive Digital Filters,” *IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, Victoria, 19-21 May 1993, pp. 232-235.
- [20] Matlab, “The Matworks™,” Ver. (R2009b).