

# Low order model for 2-D digital IIR filters

Lahcène MITICHE\*, ([l\\_mitiche@yahoo.fr](mailto:l_mitiche@yahoo.fr))

Amel Baha Houda ADAMOUMITICHE\*, ([amelmitiche@yahoo.fr](mailto:amelmitiche@yahoo.fr))

(\*) Science and Technology Department, University of Djelfa, Algeria

**Abstract**— The work presented in this paper concerns with 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. The order reduction of the filter is based on the Quasi-Gramians method.

**Keywords**— Stability, Recursively computable, Linear phase, State space, Balanced realization, Model reduction.

## I. 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 every indication that this trend will continue. Advances in hardware technology provide the capability in signal processing chips and microprocessors which were previously associated with mainframe computers. These advances allow sophisticated signal processing 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. For 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 filter's 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. 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, because poles are surfaces rather than points and there is no fundamental theorem of algebra, it is extremely difficult to determine the pole locations [1].

The work presented here is relates more particularly to the analysis and synthesis of the two-dimensional infinite impulse response filters of a reduced order. two parts are made, the synthesis (design) in the first part is complete in both areas, spatial with two methods, the Prony's method (modified Prony) and Iterative method, and iterative Semi-Definite Programming (SDP) in the frequency domain. In the second part of this work we make an order reduction of the filter synthesized (designed) by the method of Quasi-Gramians.

## II. DESIGNING A IIR FILTERS

An IIR filter with an arbitrary impulse response  $h(n_1, n_2)$  cannot be realized since computing each output sample requires a large number of arithmetic operations [1]. 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.

### A. THE DESIGN PROBLEM

The problem of designing an IIR filter is to determine a rational and stable  $H(z_1, z_2)$  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 region of support of  $a(k_1, k_2)$  except the origin  $(0, 0)$ , and  $R_b$  represents the region of support 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  $a(n_1, n_2)$  and  $b(n_1, n_2)$  are the filter coefficients.

The first step in the design of an IIR filter is usually an initial determination of  $R_a$  and  $R_b$ , the regions of support of  $a(n_1, n_2)$  and  $b(n_1, n_2)$ . 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 region of support as  $h_d(n_1, n_2)$ .

Another consideration relates to the filter specification parameters. In low-pass filter design, for example, a small  $\delta_p$ ,  $\delta_s$  (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 specification for a particular design algorithm, and an iterative procedure may become necessary [1].

One major difference between IIR and FIR filters is on issues related to stability. An 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. Although 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 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. Simple methods do not exist for the design of 2-D analog filters. The second approach, designing an IIR filter directly in the discrete domain, can be classified into two categories. The first is the spatial domain design approach, where filters are designed by using an error criterion in the spatial domain. The second is the frequency domain design approach, 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].

## B. THE STABILITY PROBLEM

In the 1-D case, the problem of 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].

## C. SPATIAL DOMAIN DESIGN

The input often used in IIR filter design is  $\delta(n_1, n_2)$ , and the desired impulse response that is 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  $a(n_1, n_2)$  and  $b(n_1, n_2)$  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 criteria used in the filter design is

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

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

and  $R_e$  is the region of support of the error sequence. Ideally,  $R_e$  is 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 system parameters.

Minimizing the Error in (3) with respect to  $a(n_1, n_2)$  and  $b(n_1, n_2)$  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

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

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

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

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

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

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) :

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

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.

*Prony's method*

In Prony's method, the error expression minimized is

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

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

$$\text{Error} = E_1 + E_2 \quad (8a)$$

$$\text{where } E_1 = \sum_{(n_1, n_2) \in R_b} e_M^2(n_1, n_2) \quad (8b)$$

$$\text{and } E_2 = \sum_{(n_1, n_2) \notin R_b} e_M^2(n_1, n_2) \quad (8c)$$

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

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

where

$$r(k_1, k_2; l_1, l_2) = \sum_{(n_1, n_2) \notin R_b} h_d(n_1 - k_1, n_2 - k_2) \times h_d(n_1 - l_1, n_2 - l_2) \quad (9b)$$

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 [1].

*Iterative Algorithm*

It is an extension of a 1-D system identification method developed by Steiglitz and McBride [1].

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 (8) and (11)

$$e(n_1, n_2) = v(n_1, n_2) * e_M(n_1, n_2) = v(n_1, n_2) * (a(n_1, n_2) * h_d(n_1, n_2) - b(n_1, n_2)) \quad (12)$$

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  $b(n_1, n_2)$ .

*Zero-Phase Filter Design*

One characteristic of a zero-phase filter is its tendency to preserve the shape of the signal component in the passband 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  $h(-n_1, -n_2)$ . An IIR filter requires an infinite extent  $h(n_1, n_2)$ , and output mask to have 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

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

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

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

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

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

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

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),

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

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

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

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

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

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  $h_d^I(n_1, n_2)$ ,  $h_d^{II}(n_1, n_2)$ ,  $h_d^{III}(n_1, n_2)$  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

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

$H(z_1, z_2)$  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.

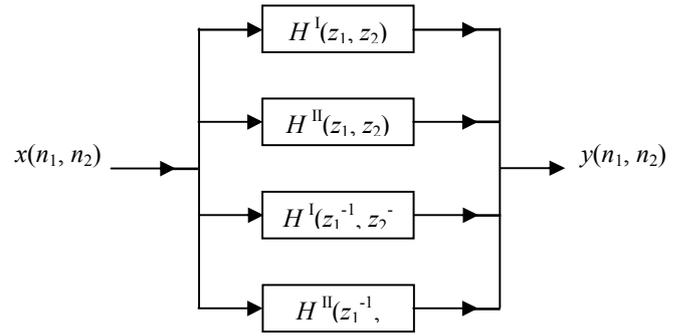


Fig. 1. Implementation of  $H(z_1, z_2)$  using a parallel structure.

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

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

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

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

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

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

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

#### D. 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 various types of FIR digital filters. An attempt on extending the SDP approach to 2-D IIR filters [2]. 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 quadrantly 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 [3].

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

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

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  $r + (n + 1)(n + 2)/2$  variables in (23), which form a  $[r + 0.5(n + 1)(n + 2)]$ -dimension vector

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

Denote vector  $x$  in the  $k^{\text{th}}$  iteration as  $x_k$  and the frequency response of the filter for  $x = x_k$  as  $H(e^{jw_1}, e^{jw_2}, x_k)$ . In the vicinity of  $x_k$ , the design variable can be expressed as

$$x = x_k + \delta$$

The transfer function can be approximated in terms of linear function of  $\delta$  by

$$H(e^{jw_1}, e^{jw_2}, x) \approx H(e^{jw_1}, e^{jw_2}, x_k) + g_k^T \delta \quad (25)$$

where  $g_k$  is the gradient of de  $H(e^{jw_1}, e^{jw_2}, x)$  for  $x = x_k$ .

**Problem Formulation**

Design in the sense of min-max is obtained as solution of the following optimization problem:

$$\text{minimize } \hat{c}^T \hat{\delta} \quad (26)$$

$$\text{subject to : } \begin{bmatrix} S_k & 0 \\ 0 & Y_k \end{bmatrix} \succeq 0 \quad (27)$$

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

where  $\mu$  is treated as an additional design variable, and

$$S_k = \text{diag}\{\Phi_k(w_1^{(1)}, w_2^{(1)}), \dots, \Phi_k(w_1^{(L)}, w_2^{(M)})\} \quad (29)$$

$$\Phi_k(w_1, w_2) = \begin{bmatrix} I & Q_k^{\frac{1}{2}} \delta \\ \delta^T Q_k^{\frac{1}{2}} & \mu - 2\delta^T q_k - c_k \end{bmatrix} \succeq 0 \quad (30)$$

$$Q_k = W(w_1, w_2) \Re e(\bar{g}_k g_k^T) \quad (31)$$

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

$$q_k = W(w_1, w_2) \times \Re e\left\{ \left[ H(e^{jw_1}, e^{jw_2}, x_k) - H_d(w_1, w_2) \right] \bar{g}_k \right\} \quad (32)$$

$$c_k = e(w_1, w_2, x_k) \quad (33)$$

$H_d(w_1, w_2)$  is the desired frequency response, for  $(w_1, w_2) \in \Omega$ , where  $\Omega = \{(w_1, w_2) : -\pi \leq w_1, w_2 \leq \pi\}$ , 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  $\tau$  is a positive scalar that stability margin of the filter.

and  $D_k = \begin{bmatrix} -(a_k + \delta_1)^T \\ \hat{I}_r \end{bmatrix}$  (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 [4], one conclude that matrix  $D_k$  is stable if and only if there exist positive definite matrix  $P$  such that:

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

Where  $M \succ 0$  denote that matrix  $M$  is positive definite. The matrix  $P$  in (34) is not considered as 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  $\hat{D}_k = \begin{bmatrix} -a_k^T \\ \hat{I}_r \end{bmatrix}$  (38)

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

**Design steps**

Give the order of the IIR filter  $(n, r)$  and the desired frequency response  $H_d(w_1, w_2)$ .

**Step 1:** The proposed design method starts with an initial point  $x_0$  that corresponds to a stable filter obtained following using a conventional method. For example, one can design an FIR filter of order  $n$  approximate  $H_d(w_1, w_2)$  and simply set

$A(z_1) \equiv 1$  and  $A(z_2) \equiv 1$  as an initial design.

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

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

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

III. 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 less 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 provides the choice of using one that is better suited for the purpose at hand [5].

A. STATE-SPACE MODEL

Roesser's model is the following [6]:

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

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

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

$$y(i, j) = \begin{bmatrix} C_1 & C_2 \end{bmatrix} \begin{bmatrix} x^h(i, j) \\ x^v(i, j) \end{bmatrix} + du(i, j), \quad (i, j) \geq (0, 0) \quad (39c)$$

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 write 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  $z_2^{-1}$ .

B. MINIMAL REALIZATION IN STATE-SPACE

The minimal state-space realizations are not always possible for all 2-D transfer functions [7]. However, minimal state-space realizations have been determined for a system with separable denominator [8].

Consider the linear time invariant 2-D system, described by the spatial transfer function [9]:

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

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  $C_2$  have 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 by:

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

$$A_2 = \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & & \vdots \\ 0 & \cdots & 0 \end{bmatrix}, A_3 = \begin{bmatrix} h(1,1) & \cdots & h(1,n) \\ \vdots & & \vdots \\ h(m,1) & \cdots & h(m,n) \end{bmatrix}$$

where

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

$$\text{and } A_4 = \begin{bmatrix} -a(1,0) & 1 & 0 & \cdots & 0 \\ \cdot & 0 & \ddots & \ddots & \cdot \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \cdot & \cdot & & \ddots & 0 \\ \cdot & \cdot & & \ddots & 1 \\ -a(m,0) & 0 & \cdot & \cdots & 0 \end{bmatrix}, B_1 = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

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

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

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

C. REDUCTION METHODS

The most popular 1-D model reduction techniques are based on the concept of balanced realization which was originally proposed by Moore [12]. 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 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 given 2-D discrete system, leading to different balanced approximations [13].

### Three types of Gramians for 2-D discrete systems

Consider the Givone-Roesser state-space model of (SISO) described in (39), we defined [13]

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

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

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 [14], 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 [15], 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 lead to the non-uniqueness 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 [16], 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:

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

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

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

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

In terms of computation complexity, the structured Gramians are the most expensive to evaluate while the quasi-Gramians are the most economical [13].

### Balanced approximations

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 [17] such as the realization characterized by  $\{T^1 A T, T^1 B, B T, d\}$  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

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

where

$$A_{1r} = A_1(1: r_1, 1: r_1), \quad A_{2r} = A_2(1: r_1, 1: r_2), \quad A_{3r} = A_3(1: r_2, 1: r_1),$$

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

$$B_{1r} = B_1(1: r_1), \quad B_{2r} = B_2(1: r_2), \quad C_{1r} = C_1(1: r_1), \quad C_{2r} = C_2(1: r_2)$$

[13].

### Iterative Algorithm for the case Quasi-Gramians

An iterative method for the computation of Quasi-Gramians is described, where each iteration involves solving two 1-D Lyapunov equations. For 2-D system stable, the algorithm converges to the 2-D quasi-Gramians very quickly [18].

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

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

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

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

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  $P_2^{(k)}$  et  $Q_2^{(k)}$ :

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

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

where  $F_2 = B_2 B_2^T + A_3 P_1^{(k)} A_3^T$   
 $G_2 = C_2^T C_2 + A_2^T Q_1^{(k)} A_2$

**Step 4:** Set  $k=k+1$  and repeat Step 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  $\varepsilon$  is a prescribed tolerance [18].

## IV. A DESIGN EXAMPLES

We have divide the simulation into two parts:

### Part 1: the Design

The design have been finished in two domains:

- 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 function

Impulse\_2D.m to produce the impulse response and frequency response.

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

For the Prony's method and Iterative method the IIR digital filters given by the following templates:

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

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

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

Bandpass:  $R_p = [0.4 \quad 0.6]$ ,  $R_{s1} = [0 \quad 0.3]$  and  $R_{s2} = [0.7 \quad 1]$ .

**Prony's method**

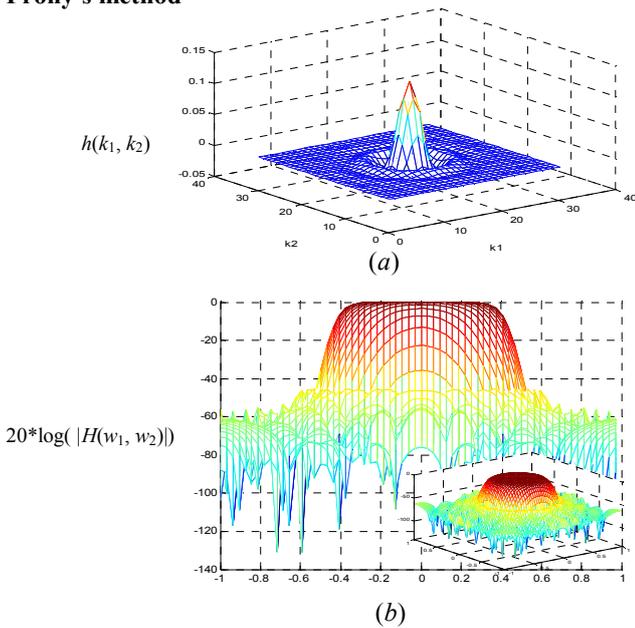


Fig. 2 . Zero-phase lowpass IIR filter. *a*) Impulse response. *b*) : Amplitude responses (in decibels),"X-Z" perspective.

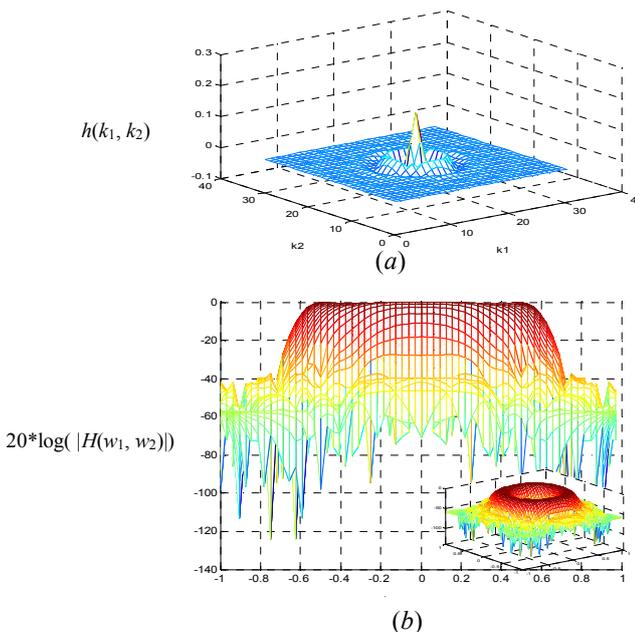


Fig. 3. Zero-phase bandpass IIR filter. *a*) Impulse response. *b*) : Amplitude responses (in decibels),"X-Z" perspective.

**Iterative method**

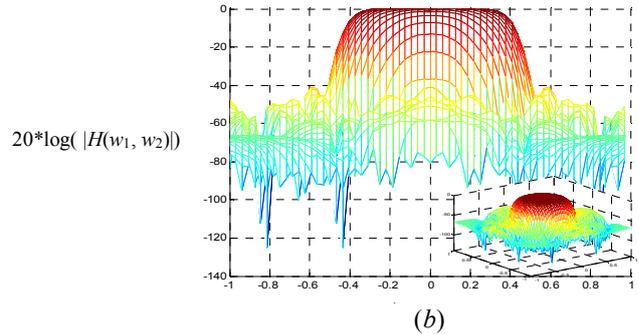
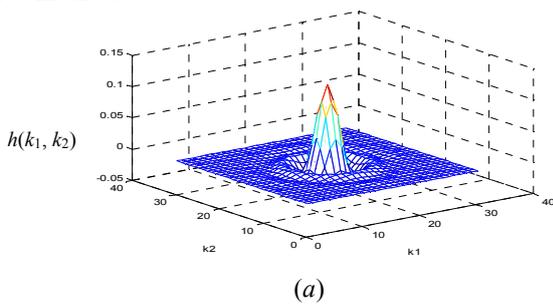


Fig. 4. Zero-phase lowpass IIR filter. *a*) Impulse response. *b*) : Amplitude responses (in decibels),"X-Z" perspective.

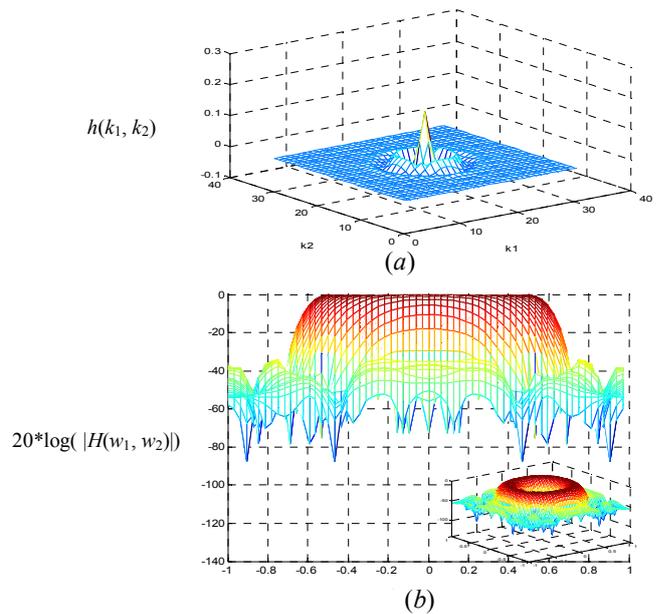
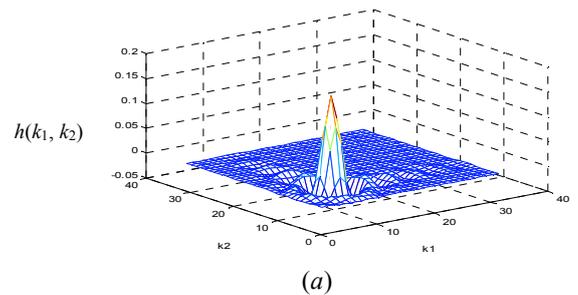


Fig. 5. Zero-phase bandpass IIR filter. *a*) Impulse response. *b*) : Amplitude responses (in decibels),"X-Z" perspective.

For the SDP method: Let IIR digital filters given by the following templates:  
 Filter order  $n = 13$ ,  $r = 13$  (the  $a$  and  $b$  matrix dimension).  
 The passband and stopband corresponds 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 :  $it = 2$ .

**SDP method**



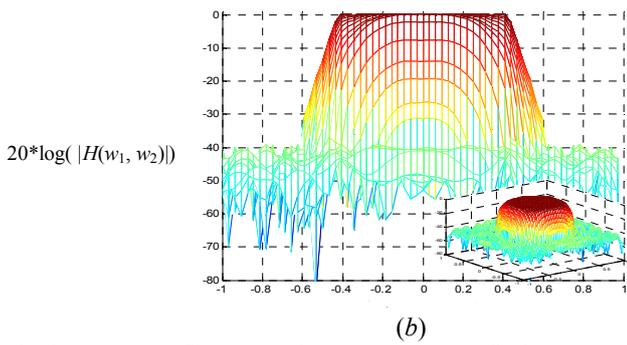


Fig. 6. Lowpass IIR filter. a) Impulse response. b) : Amplitude responses (in decibels),"X-Z" perspective.

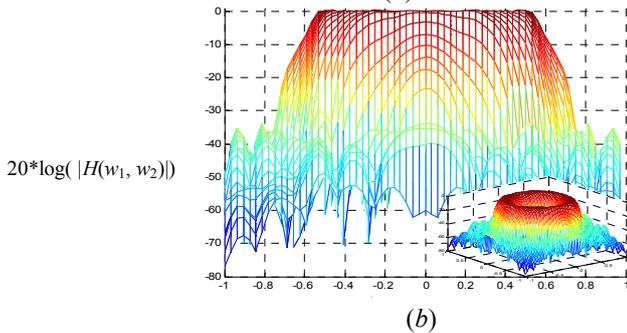
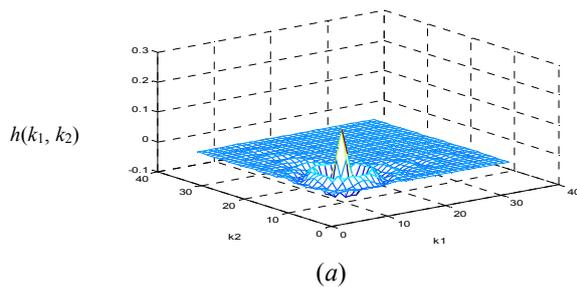


Fig. 7. Bandpass IIR filter. a) Impulse response. b) : Amplitude responses (in decibels),"X-Z" perspective.

Part 2: the Reduction

We apply the method of Quasi-Gramians to lowpass filter designed in the first part, we use our function `tf2ss2_2D.m` to transfer matrices  $a$  and  $b$  to the state-space model  $(A, B, C, d)$ , then applied Quasi-Gramians method to produce a reduced-order model  $(A_r, B_r, C_r, d)$ .

**Iterative method**

The order of original filter is:  $n = 5, m = 5$ .  
 - the total number of coefficient is:  $(5 \times 5$  (matrix  $b$ )  $+ 5 \times 5$  (matrix  $a$ ))  $\times 4 = 200$ .  
 The dimension of the matrix  $A, B,$  and  $C$  are  $(n + 2 \times m) \times (n + 2 \times m), (n + 2 \times m) \times 1, 1 \times (n + 2 \times m)$ , respectively.  
 Number of iterations is:  $it = 10$ .  
 Order of reduced filter is:  $r_1 = 4, r_2 = 5$ .  
 - the total number of coefficient is  $(4 \times 5$  (matrix  $b$ )  $+ 4 \times 5$  (matrix  $a$ ))  $\times 4 = 160$ .

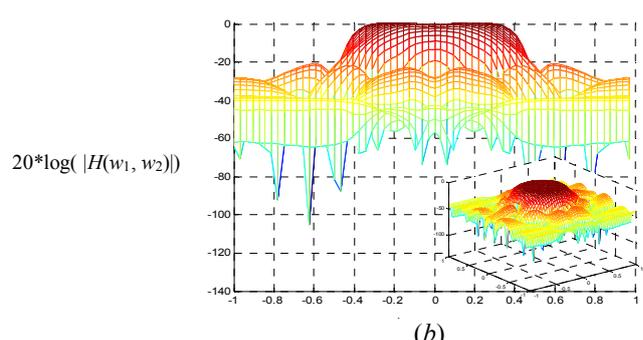
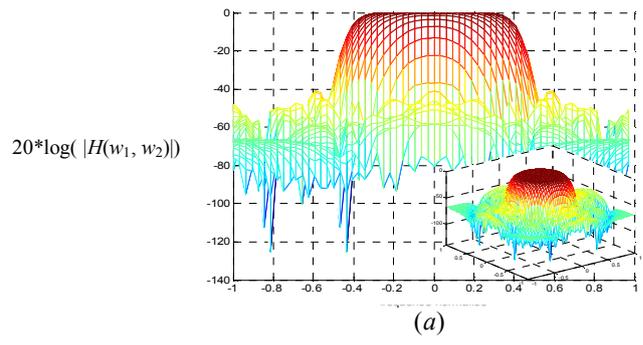


Fig. 8. Lowpass IIR filter amplitude responses (in decibels),"X-Z" perspective. a) Original filter  $5 \times 5$  (200 coefficients). b) Reduced-order filter  $4 \times 5$ , (160 coefficients).

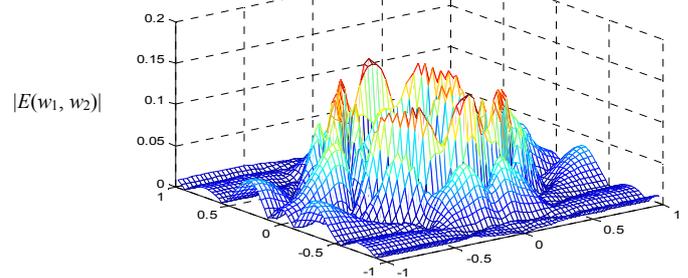
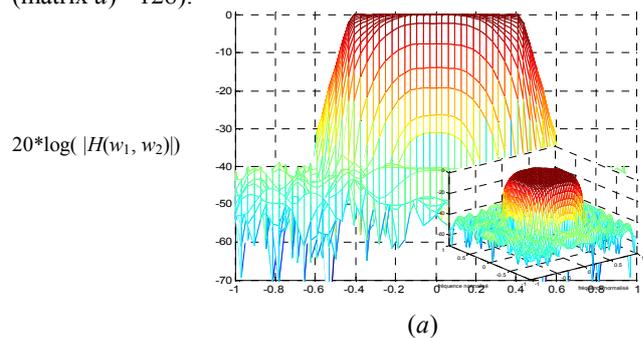


Fig. 9. Error specter between original and reduced filter.

**SDP method**

The order of original filter is:  $n = 13, m = 13$ .  
 - the total number of coefficient is:  $13 \times 13$  (matrix  $b$ )  $+ 13 \times 13$  (matrix  $a$ )  $= 338$ .  
 The dimension of the matrix  $A, B,$  and  $C$  are  $(n + m) \times (n + m), (n + m) \times 1, 1 \times (n + m)$ , 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:  $8 \times 8$  (matrix  $b$ )  $+ 8 \times 8$  (matrix  $a$ )  $= 128$ .



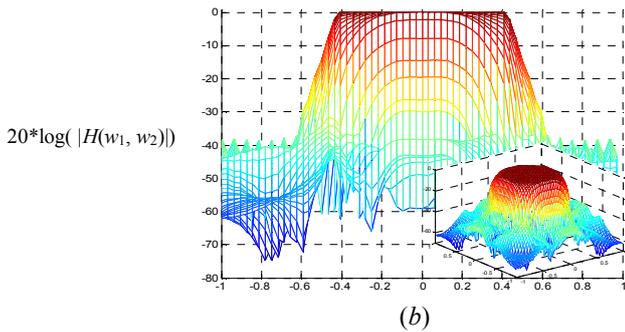


Fig. 10. Lowpass IIR filter amplitude responses (in decibels), "X-Z" perspective. a) Original filter  $13 \times 13$  (338 coefficients). b) Reduced-order filter  $8 \times 8$ , (128 coefficients).

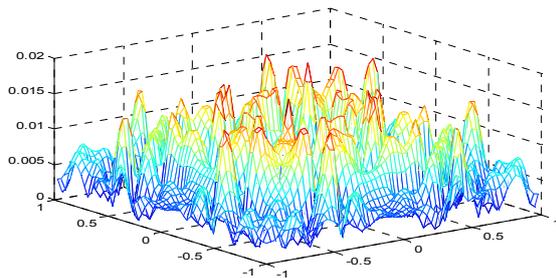


Fig. 11. Error specter between original and reduced filter.

## V. RESULT AND INTERPRETATION

The results obtained lead to the following:

### \* 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 applied :

Prony / min ( $H(w_1, w_2)$ ) = -49.09 dB (lowpass filter).

Iterative / min ( $H(w_1, w_2)$ ) = -55.77 dB (lowpass filter).

- The results obtained in the SDP method are comparable to the other two methods, we found a performance decreasing in both bands (passband and stopband). It is possible to improve results by increasing number of coefficients ( $n, r$ ) or number of iterations. There is a good aspect of this method is the stability, which can be measured by the max of roots of vector of column or row of the denominator matrix 'a', we found the following results:

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

where max, abs, and roots are functions in Matlab, which represents the maximum value of a vector, the absolute value, and the roots of a vector, respectively.

### \* The reduction:

- Due to the non-minimal realization of filter designed by Prony and Iterative methods, and because the reduced filter can be unstable, therefore the reduction results are not good (see fig 8, 9).

- But in the SDP method the stability is preserved:

$\max(\text{abs}(\text{roots}(a(:, 1)))) = 0.8926 \Rightarrow$  (original filter)  $\Rightarrow$   
 $0.9119 \Rightarrow$  (filter reduces) for low-pass filter.

The results obtained apparent that the reduced low-pass filter is acceptable as the number of coefficients decrease from 338

to 128, and the max error between the reduced and original filter is :  $\max(E) = 0.02$  (fig 11) .

## VI. CONCLUSION

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

The reduction also suffer from some problems, reduction in state-space by conventional methods, for example, reduction by the balanced transformation require a minimal realization of a transfer function in state-space, then find a method to compute the transformation matrix to a balanced model, which requires that as small number as possible of calculation and at the same time maintaining the stability.

- In this work we attempt to find a design method that solve these problems with conventional methods such as Prony's method or Iterative method where they based on identification idea. For a zero-phase filter, we divided the support region of the desired impulse response on four regions, then we have synthesized the four filters (one filter in the case where  $h_d(n_1, n_2)$  have a four-fold symmetry) by the Prony method or Iterative method. The result is a linear phase filter has minimal order (total number of coefficients equal 200) compared by the third method. In the method for Semi-definite Programming (total number of coefficients equal 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] Jae S.Lim, Two-Dimensional Signal and Image Processing, *Department of Electrical Engineering and Computer Science MIT*, Prentice Hall PTR1990.
- [2] W.-S. Lu, "Design of stable 2-D IIR digital filters using iterative Semidefinite programming," in *Proc. ISCAS'00*, Geneva, Switzerland, May 2000.
- [3] 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, pp. 814-826, June 2002.
- [4] T. Kailath, *Linear Systems*, Prentice-Hall, 1981.
- [5] K. Premaratne, E. I. Jury, and M. Mansour, "An algorithm for model reduction of 2-D discrete time systems," *IEEE Trans. Circuits Syst*, vol. 38, pp. 1116-1132, Sept. 1990.
- [6] 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," *Proc. IEEE*, vol. 65, pp. 945-961, June 1977.

[7] E. D. Sontag, "On first order equations for multidimensional filters," *IEEE Trans. Automat. Speech, Signal Processing*, vol. ASSP-26, pp. 480-482, Oct. 1978.

[8] S. Attasi, "Modélisation et traitement des suites à deux indices," *IRIA, Rapport Lab*, Sept. 1975.

[9] J.L. Shanks, S. Treitel, and J.H. Justice, "Stability and synthesis of tow-dimensional recursive filters," *IEEE Trans. Audio Electroacoust.*, vol. AU-20, pp. 115-128, June. 1972.

[10] D. D. Givone and R. P. Roesser, "Minimization of multidimensional linear iterative circuits," *IEEE Trans. Comput.*, vol. C-22, pp. 673-678, July 1973.

[11] G.E. Antoniou, P.N. Paraskevopoulos, and S.J. Varoufakis, "Minimal State-Space Realization of Factorable 2-D Transfer Function," *IEEE Trans. Circuits Syst.*, vol. 35, pp. 1055-1058, Aug. 1988.

[12] B. C. Moore, "Principal component analysis in linear systems: Controllability, observability and model reduction", *IEEE Trans. Automat. Contr.*, vol. 26, pp. 17-32, Feb. 1981.

[13] W. S. Lu, H. Luo and A. Antonio, "Recent Results on Model Reduction Methods for 2-D Discrete Systems", *IEEE*, pp. 348-351, 1996.

[14] W.-S. Lu, E. B. Lee, and Q. T. Zhang, "Balanced approximation of two-dimensional and delay differential systems", *Int. J. Contr.*, vol. 47, pp. 2199-2218, 1987.

[15] W. Wang, K. Glover, J. C. Doyle, and C. Beck, "Model reduction of LFT systems", *Proc. 30th IEEE Conf. on Decision and Control*, pp. 1233-1238, Brighton, UK, Dec. 1991.

[16] K. Zhou, J. L. Aravena, G. Gu, and D. Xiong, "Two-dimensional system model reduction by quasi-balanced truncation and singular perturbation", *IEEE Trans. Circuits Syst.*, vol. 41, pp. 593-602, Sept. 1994.

[17] A. J. Laub, "On computing balancing transformations", *Proc. JACC*, FA 8-E, San Francisco, 1980.

[18] 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," in *Proc. IEEE Pacific Rim Con& Commun., Comput., Signal Process.*, Victoria, B.C., Canada, pp. 232-235, May 1993.

**Dr Lahcène MITICHE** received the state engineering, the Master of sciences, and the Ph. D. Degrees all in Electronics (Signal processing), from Ecole Nationale Polytechnique of Algiers, Algeria, in 1991, 1994, and 2005 respectively, with full distinction. Recently, he is head of Sciences and Technology Department and Associate professor in signal processing, of Djelfa University (Djelfa, Algeria). His research interests include model reduction with several applications to large scale analog and digital, scalar and 2D systems, speech and image signals.

**Dr Amel Baha Houda ADAMOUMITICHE** received the state engineering, the Master of sciences, and the Ph. D. Degrees all in Electronics (Control systems), from Ecole Nationale Polytechnique of Algiers, Algeria, in 1993, 1996, and 2006 respectively, with full distinction. Recently, he is head of Sciences and Technology Department Scientific

Committee and Associate professor in Control systems, of Djelfa University (Djelfa, Algeria). His research interests include model reduction with several applications to large scale analog and digital, scalar and 2D systems, generalized state space models and descriptor systems approximation.