

© 2007 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

# On the realisation of 2-D linear systems with implicit latent variable models

Ran Yang, Lorenzo Ntogramatzidis and Michael Cantoni

**Abstract**—In this paper a new latent variable model is proposed for the realisation of 2-D linear causal systems. For the general  $(n, m)$ -th order 2-D system, the latent variable model obtained here is implicit when  $n \neq m$ . Importantly however, even in this case the model is recursively computable. The advantage of this realisation is that its order is  $nm$ , which is much smaller than the orders of the existing latent variable realisations such as Fornasini-Marchesini (FM) models and Roesser model. In addition, given a particular 2-D system, an algorithm is developed for further reducing the size of the realisation.

## I. INTRODUCTION

In the past three decades, two-dimensional (2-D) latent variable models have been widely used in the analysis and synthesis of 2-D linear systems, including stability analysis [19], [4], digital filter design [10], [4], estimation [25], [5], optimal control [5], [18]. Moreover, 2-D latent variable models have been successfully applied to the design of iterative learning control (ILC) schemes [16], [1].

A motivation for the use of latent variable representations of causal input-output processes – both within a one and two-dimensional context – lies in the possibility of computing the output of a system recursively, given the input. In fact, the recursive structure is often exploited when dealing with optimal synthesis problems (e.g. dynamic programming). As such, one fundamental issue of systems and control theory is the realisation of input-output transfer function representations in terms of latent variable models. While in the one dimensional case the latent variable model usually employed to this purpose is the classic state-space model, in the 2-D case there are more degrees of freedom in the choice of the type of the latent variable model to be employed. In fact, the models commonly utilised for the realisation of causal 2-D transfer functions are the Roesser model [23], Fornasini-Marchesini first (FM-I) model [6] and second (FM-II) model [7], the Kurek model [15].

When the input-output process to be realised is not causal, a realisation can be obtained in terms of so-called *implicit* models. These models were first introduced in [20] and [21] in the 1-D case, and then extended to the multidimensional

This work was supported in part by The University of Melbourne (MRGS), the Australian Research Council (Discovery Project DP0664789), National Natural Science Foundation of China (Grant 60504022), Natural Science Foundation of Guangdong (Grant 05003343).

R. Yang is with the School of Information Science and Technology, Sun Yat-Sen University, Guangzhou 510275, P.R. China. yangran@mail.sysu.edu.cn

L. Ntogramatzidis and M. Cantoni are with the Department of Electrical and Electronic Engineering, The University of Melbourne, Parkville VIC 3010, Australia {lnt, m.cantoni}@ee.unimelb.edu.au

TABLE I  
COMPARISON OF THE SIZES OF DIFFERENT REALISATIONS FOR (1)

model type	Roesser	FM-I	FM-II	FM-II
realisation given in	[11]	[6]	[3]	[24]
size of realisation	6	16	14	7

case in [12] and [17] for Fornasini-Marchesini and Roesser models, respectively. The general drawback of implicit models is their lack of recursibility. In other words, for implicit latent variable models an iterative technique for the solution of the associated one or two-dimensional difference equation may not exist, [17].

In this paper, we show that *causal* 2-D transfer functions can be realised in terms of implicit latent variable models without losing the property of recursive computability of the input-output relation given appropriate boundary conditions. By taking advantage of implicit models for the realisation of causal transfer functions, we obtain a realisation whose size is smaller than the sizes of the usual realisations given in terms of Roesser or Fornasini-Marchesini models. More precisely, for a general 2-D system described by an  $(n, m)$ -th order transfer function, a realisation with system matrices of size  $nm$  is provided. It is also shown that the size of this realisation may be further reduced if some coefficients appearing in the transfer function are zero. And for this purpose, an algorithm is developed for the reduction of the size of the realisation in these particular cases.

The motivation for developing the realisation technique presented here with implicit models is shown by an example.

**Example 1:** Consider the 2-D sequences  $u_{i,j}$  and  $y_{i,j}$  which are representative of the input and the output of the 2-D  $(1, 3)$ -th order process ruled by

$$\left( \sum_{i=0}^1 \sum_{j=0}^3 a_{i,j} \lambda_h^i \lambda_v^j \right) Y(\lambda_h, \lambda_v) = \left( \sum_{i=0}^1 \sum_{j=0}^3 b_{i,j} \lambda_h^i \lambda_v^j \right) U(\lambda_h, \lambda_v) \\ a_{0,0} = 1, \quad b_{0,0} = 0, \quad (1)$$

where  $U(\lambda_h, \lambda_v)$  and  $Y(\lambda_h, \lambda_v)$  are formal power series associated with the two sequences  $u_{i,j}$  and  $y_{i,j}$  by means of the transformation  $S(\lambda_h, \lambda_v) := \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} s_{i,j} \lambda_h^i \lambda_v^j$  that maps the bivariate sequence  $s_{i,j}$  into the formal power series  $S(\lambda_h, \lambda_v)$ . The coefficients  $a_{i,j}, b_{i,j}$  are real. The sizes of the realisations by different models and with different techniques are shown in Table I.

Consider the following latent variable model

$$\begin{aligned} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x'_{i+1,j+1} \\ x''_{i+1,j+1} \end{bmatrix} &= \begin{bmatrix} 0 & 0 & -a_{1,3} \\ 1 & 0 & -a_{1,2} \\ 0 & 1 & -a_{1,1} \end{bmatrix} \begin{bmatrix} x'_{i,j} \\ x''_{i,j} \end{bmatrix} \\ + \begin{bmatrix} 0 & 0 & -a_{0,3} \\ 0 & 0 & -a_{0,2} \\ 0 & 0 & -a_{0,1} \end{bmatrix} \begin{bmatrix} x'_{i+1,j} \\ x''_{i+1,j} \end{bmatrix} &+ \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -a_{1,0} \end{bmatrix} \begin{bmatrix} x'_{i,j+1} \\ x''_{i,j+1} \end{bmatrix} \\ + \begin{bmatrix} b_{1,3} \\ b_{1,2} \\ b_{1,1} \end{bmatrix} u_{i,j} + \begin{bmatrix} b_{0,3} \\ b_{0,2} \\ b_{0,1} \end{bmatrix} u_{i+1,j} &+ \begin{bmatrix} 0 \\ 0 \\ b_{1,0} \end{bmatrix} u_{i,j+1} \end{aligned} \quad (2)$$

$$y_{i,j} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x'_{i,j} \\ x''_{i,j} \end{bmatrix} \quad (3)$$

along with the boundary conditions  $x'_{i,j} = 0$  for all  $(i,j) \in [-1,\infty) \times \{-1\}$  and  $x''_{i,j} = 0, u_{i,j} = 0$  for all  $(i,j) \in [-1,\infty) \times \{-1\} \cup \{-1\} \times [-1,\infty)$ . Notice that this model is implicit. However, with these boundary conditions it is still causal in terms of the input-output relationship, since the first equation can be written explicitly as

$$z_{i,j} = - \sum_{k=0}^1 \sum_{l=2}^3 a_{k,l} x''_{i+1-k,j+1-l} + \sum_{k=0}^1 \sum_{l=2}^3 b_{k,l} u_{i+1-k,j+1-l} \quad (4)$$

$$\begin{aligned} x''_{i+1,j+1} &= z_{i,j} + \sum_{(k,l) \in \{(0,1), (1,0), (1,1)\}} (-a_{k,l} x''_{i+1-k,j+1-l} \\ &+ b_{k,l} u_{i+1-k,j+1-l}). \end{aligned} \quad (5)$$

where  $z_{i,j} := [0 \ 1] x'_{i,j}$ . By substitution of (4) into (5), and by taking the boundary conditions into account, the input-output relation of the model thus obtained is found to be consistent with (1). In other words, the latent variable model obtained is a realisation of (1) and its size is 3, which is much smaller than the orders of the realisations shown in Table I.

Observe that if  $z_{i,j}$  is removed from (5), together with the output equation, (5) represents a (1,1)-st order 2-D system, i.e.

$$\begin{aligned} (1+a_{1,0}\lambda_h+a_{0,1}\lambda_v+a_{1,1}\lambda_h\lambda_v)Y(\lambda_h, \lambda_v) \\ = (b_{1,0}\lambda_h+b_{0,1}\lambda_v+b_{1,1}\lambda_h\lambda_v)U(\lambda_h, \lambda_v). \end{aligned}$$

The addition of another latent variable  $z_{i,j}$  yields a realisation for the (1,3)-th order system (1).

Following this idea, a new realisation method can be developed for the general  $(n,m)$ -th order 2-D system

$$\left( \sum_{i=0}^n \sum_{j=0}^m a_{i,j} \lambda_h^i \lambda_v^j \right) Y(\lambda_h, \lambda_v) = \left( \sum_{i=0}^n \sum_{j=0}^m b_{i,j} \lambda_h^i \lambda_v^j \right) U(\lambda_h, \lambda_v) \quad (6)$$

with  $a_{0,0} = 1$ , where  $n$  is not necessarily equal to  $m$ . The latent variable model used here is of the implicit form

$$\begin{aligned} E x_{i+1,j+1} &= A_0 x_{i,j} + A_1 x_{i+1,j} + A_2 x_{i,j+1} \\ &+ B_0 u_{i,j} + B_1 u_{i+1,j} + B_2 u_{i,j+1} \end{aligned} \quad (7)$$

$$y_{i,j} = C x_{i,j} + D u_{i,j}$$

with the following boundary conditions

$$x_{i,j} = 0 \text{ and } u_{i,j} = 0 \quad \forall (i,j) \in \mathbb{B} \quad (8)$$

where  $\mathbb{B} := [-1,\infty) \times \{-1\} \cup \{-1\} \times [-1,\infty)$ . In Section II it is shown that the solution of (7) with boundary conditions (8) is uniquely determined, and can be computed recursively. When  $n = m$ ,  $E$  is an identity matrix. When  $n < m$ , the matrices of the model (7) have the following structure

$$\begin{aligned} E &= \begin{bmatrix} 0 & 0 \\ 0 & I \end{bmatrix}, A_0 = \begin{bmatrix} A_0^{22} & A_0^{21} \\ A_0^{12} & A_0^{11} \end{bmatrix}, A_1 = \begin{bmatrix} 0 & A_1^{21} \\ 0 & A_1^{11} \end{bmatrix}, \\ A_2 &= \begin{bmatrix} -I & 0 \\ 0 & A_2^{11} \end{bmatrix}, B_0 = \begin{bmatrix} B_0^2 \\ B_0^1 \end{bmatrix}, B_1 = \begin{bmatrix} B_1^2 \\ B_1^1 \end{bmatrix}, B_2 = \begin{bmatrix} 0 \\ B_2^1 \end{bmatrix}. \end{aligned} \quad (9)$$

When  $n > m$ , the matrices of the latent variable model (7) have the following structure

$$\begin{aligned} E &= \begin{bmatrix} 0 & 0 \\ 0 & I \end{bmatrix}, A_0 = \begin{bmatrix} A_0^{22} & A_0^{21} \\ A_0^{12} & A_0^{11} \end{bmatrix}, A_1 = \begin{bmatrix} -I & 0 \\ 0 & A_1^{11} \end{bmatrix}, \\ A_2 &= \begin{bmatrix} 0 & A_2^{21} \\ 0 & A_2^{11} \end{bmatrix}, B_0 = \begin{bmatrix} B_0^2 \\ B_0^1 \end{bmatrix}, B_1 = \begin{bmatrix} 0 \\ B_1^1 \end{bmatrix}, B_2 = \begin{bmatrix} B_2^1 \\ B_2^2 \end{bmatrix}. \end{aligned} \quad (10)$$

The details about the realisation for this model are given in Section II. Based on this, an algorithm is then developed to reduce the size of the realisation for 2-D systems with particular zero structure.

## II. MAIN RESULTS

The solution to the realisation for the 2-D general model is given in the next theorem.

*Theorem 1:* Given an  $(n,m)$ -th order 2-D system in the form of (6), a realisation in terms of the model (7-8) is given by the following matrices:

$$A_k = \begin{bmatrix} A_k^{(r,r)} & \mathbf{0} & \cdots & \mathbf{0} & A_k^{(r,1)} \\ A_k^{(r-1,r)} & \ddots & \ddots & \vdots & \vdots \\ \mathbf{0} & \ddots & \ddots & \mathbf{0} & \vdots \\ \vdots & \ddots & A_k^{(2,3)} & A_k^{(2,2)} & A_k^{(2,1)} \\ \mathbf{0} & \cdots & \mathbf{0} & A_k^{(1,2)} & A_k^{(1,1)} \end{bmatrix}, B_k = \begin{bmatrix} B_k^{(r)} \\ B_k^{(r-1)} \\ \vdots \\ B_k^{(2)} \\ B_k^{(1)} \end{bmatrix},$$

for  $k = 0, 1, 2$  and

$$E = \begin{bmatrix} E^{(r,r)} & & \\ & \ddots & \\ & & E^{(1,1)} \end{bmatrix}, C = [0 \ \cdots \ 0 \ 0 \ 1],$$

with  $r = \max(n,m)$ ,  $s = \min(n,m)$ , and

$$\begin{aligned} E^{(i,i)} &= \begin{cases} I_{(2i-1) \times (2i-1)}, & i = 1, \dots, s \\ \mathbf{0}_{s \times s}, & i = s+1, \dots, r \end{cases} \\ A_0^{(i,1)} &= \begin{cases} -[a_{i,i} \ a_{i,i-1} \ a_{i-1,i} \ \cdots \ a_{i,1} \ a_{1,i}]_{1 \times (2i-1)}^T, & i = 1, \dots, s \\ -[a_{i,s} \ \cdots \ a_{i,2} \ a_{i,1}]_{1 \times s}^T, & i = s+1, \dots, r \text{ if } m < n \\ -[a_{s,i} \ \cdots \ a_{2,i} \ a_{1,i}]_{1 \times s}^T, & i = s+1, \dots, r \text{ if } m > n \end{cases} \\ A_0^{(i-1,i)} &= \begin{cases} M'_{(2i-3) \times (2i-1)}, & i = 2, \dots, s \\ M''_{(2s-1) \times s}, & i = s+1 \text{ if } n < m \\ M'''_{(2s-1) \times s}, & i = s+1 \text{ if } n > m \\ I_{s \times s}, & i = s+2, \dots, r \end{cases} \end{aligned} \quad (11)$$

with

$$M'_{(2i-3) \times (2i-1)}[k, l] = \begin{cases} 1, & k = l = 1 \\ 0, & \text{otherwise} \end{cases}$$

$$M''_{(2s-1) \times s}[k, l] = \begin{cases} 1, & k = l = 1 \\ 1, & k = 2l - 2, l > 1 \\ 0 & \text{otherwise} \end{cases}$$

$$M'''_{(2s-1) \times s}[k, l] = \begin{cases} 1, & k = l = 1 \\ 1, & k = 2l - 1, l > 1 \\ 0 & \text{otherwise} \end{cases}$$

$$A_0^{(i,i)} = \begin{cases} \mathbf{0}_{(2i-1) \times (2i-1)}, & i = 2, \dots, s \\ \mathbf{0}_{s \times s}, & i = s+1, \dots, r \end{cases}$$

$$A_1^{(i,1)} = \begin{cases} [0 \ \dots \ 0 \ -a_{0,i}]_{1 \times (2i-1)}^T, & i = 1, \dots, s \\ [0 \ \dots \ 0 \ -a_{0,i}]_{1 \times s}^T, & i = s+1, \dots, r \text{ if } m > n \\ \mathbf{0}_{s \times 1}, & i = s+1, \dots, r \text{ if } m < n \end{cases}$$

$$A_1^{(i-1,i)} = \begin{cases} N'_{(2i-3) \times (2i-1)}, & i = 2, \dots, s \\ \mathbf{0}_{(2s-1) \times s}, & i = s+1 \\ \mathbf{0}_{s \times s}, & i = s+2, \dots, r \end{cases}$$

with

$$N'_{(2i-3) \times (2i-1)} = \begin{cases} 1, & l = k+2, k \text{ is odd} \\ 0, & \text{otherwise} \end{cases}$$

$$A_1^{(i,i)} = \begin{cases} \mathbf{0}_{(2i-1) \times (2i-1)}, & i = 2, \dots, s \\ -I_{s \times s}, & i = s+1, \dots, r \text{ if } m < n \\ \mathbf{0}_{s \times s}, & i = s+1, \dots, r \text{ if } m > n \end{cases}$$

$$A_2^{(i,1)} = \begin{cases} [-a_{1,0}, \ i = 1, \\ [0 \ \dots \ 0 \ -a_{i,0} \ 0]_{1 \times (2i-1)}^T, & i = 2, \dots, s \\ [0 \ \dots \ 0 \ -a_{i,0}]_{1 \times s}^T, & i = s+1, \dots, r \text{ if } m < n \\ \mathbf{0}_{s \times 1}, & i = s+1, \dots, r \text{ if } m > n \end{cases}$$

$$A_2^{(i-1,i)} = \begin{cases} Z_{(2i-3) \times (2i-1)}, & i = 2, \dots, s \\ \mathbf{0}_{(2s-1) \times s}, & i = s+1, \\ \mathbf{0}_{s \times s}, & i = s+2, \dots, r \end{cases}$$

with

$$Z_{(2i-3) \times (2i-1)}[k, l] = \begin{cases} 1, & k = 1, l = 2 \\ 1, & l = k+2, k = 2, 4, \dots, 2i-4 \\ 0, & \text{otherwise} \end{cases}$$

$$A_2^{(i,i)} = \begin{cases} \mathbf{0}_{(2i-1) \times (2i-1)}, & i = 2, \dots, s \\ -I_{s \times s}, & i = s+1, \dots, r \text{ if } m > n \\ \mathbf{0}_{s \times s}, & i = s+1, \dots, r \text{ if } m < n \end{cases}$$

$B_0^{(i)} =$

$$\begin{cases} [\bar{b}_{i,i} \ \dots \ \bar{b}_{i,2} \ \bar{b}_{2,i} \ \bar{b}_{i,1} \ \bar{b}_{1,i}]_{1 \times (2i-1)}^T, & i = 1, \dots, s \\ [\bar{b}_{i,q} \ \dots \ \bar{b}_{i,2} \ \bar{b}_{i,1}]_{1 \times s}^T, & i = s+1, \dots, r \text{ if } m < n \\ [\bar{b}_{s,i} \ \dots \ \bar{b}_{2,i} \ \bar{b}_{1,i}]_{1 \times s}^T, & i = s+1, \dots, r \text{ if } m > n \end{cases}$$

$$B_1^{(i)} = \begin{cases} [0 \ \dots \ 0 \ \bar{b}_{0,i}]_{1 \times (2i-1)}^T, & i = 1, \dots, s \\ \mathbf{0}_{s \times 1}, & i = q+1, \dots, r \text{ if } m < n \\ [0 \ \dots \ 0 \ \bar{b}_{0,i}]_{1 \times q}^T, & i = s+1, \dots, r \text{ if } m > n \end{cases}$$

$$B_2^{(i)} = \begin{cases} \bar{b}_{1,0}, & i = 1, \\ [0 \ \dots \ 0 \ \bar{b}_{i,0} \ 0]_{1 \times (2i-1)}^T, & i = 2, \dots, s \\ [0 \ \dots \ 0 \ \bar{b}_{i,0}]_{1 \times q}^T, & i = s+1, \dots, r \text{ if } m < n \\ \mathbf{0}_{s \times 1}, & i = q+1, \dots, r \text{ if } m > n \end{cases}$$

$$D = b_{0,0},$$

where  $\bar{b}_{i,j} := b_{i,j} - a_{i,j}b_{0,0}$  for all  $i = 0, 1, \dots, n$ , and  $j = 0, 1, \dots, m$ .

**Sketch of the proof:** It can be shown by direct calculation that

$$\det(H(\lambda_h, \lambda_v)) = \begin{cases} \sum_{i=0}^n \sum_{j=0}^m a_{i,j} \lambda_h^i \lambda_v^j, & n = m \\ \lambda_h^{n(m-n)} \sum_{i=0}^n \sum_{j=0}^m a_{i,j} \lambda_h^i \lambda_v^j, & n < m \\ \lambda_v^{m(n-m)} \sum_{i=0}^n \sum_{j=0}^m a_{i,j} \lambda_h^i \lambda_v^j, & n > m \end{cases}$$

and

$$C \text{adj}(H(\lambda_h, \lambda_v)) V(\lambda_h, \lambda_v) = \begin{cases} \sum_{i=0}^n \sum_{j=0}^m b_{i,j} \lambda_h^i \lambda_v^j - D \sum_{i=0}^n \sum_{j=0}^m a_{i,j} \lambda_h^i \lambda_v^j, & n = m \\ \lambda_h^{n(m-n)} \left( \sum_{i=0}^n \sum_{j=0}^m b_{i,j} \lambda_h^i \lambda_v^j - D \sum_{i=0}^n \sum_{j=0}^m a_{i,j} \lambda_h^i \lambda_v^j \right), & n < m \\ \lambda_v^{m(n-m)} \left( \sum_{i=0}^n \sum_{j=0}^m b_{i,j} \lambda_h^i \lambda_v^j - D \sum_{i=0}^n \sum_{j=0}^m a_{i,j} \lambda_h^i \lambda_v^j \right), & n > m, \end{cases}$$

where

$$H(\lambda_h, \lambda_v) := E - A_0 \lambda_h \lambda_v - A_1 \lambda_v - A_2 \lambda_h,$$

$$V(\lambda_h, \lambda_v) := B_0 \lambda_h \lambda_v + B_1 \lambda_v + B_2 \lambda_h,$$

see [22] for the details. We want to show that in all three cases (6) is satisfied. Let  $X(\lambda_h, \lambda_v)$ ,  $U(\lambda_h, \lambda_v)$ ,  $Y(\lambda_h, \lambda_v)$  denote the formal power series associated with the signals  $x$ ,  $u$  and  $y$ . Moreover, notice that by  $\sum_{i=0}^{\infty} \sum_{j=0}^{\infty} x_{i+1,j} \lambda_h^i \lambda_v^j = \sum_{k=1}^{\infty} \sum_{j=0}^{\infty} x_{k,j} \lambda_h^{k-1} \lambda_v^j$ , we find

$$\lambda_h \left( \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} x_{i+1,j} \lambda_h^i \lambda_v^j \right) = X(\lambda_h, \lambda_v) - \sum_{j=0}^{\infty} x_{0,j} \lambda_v^j.$$

Similarly  $\lambda_v \left( \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} x_{i,j+1} \lambda_h^i \lambda_v^j \right) = X(\lambda_h, \lambda_v) - \sum_{i=0}^{\infty} x_{i,0} \lambda_h^i$

and  $\lambda_h \lambda_v \left( \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} x_{i+1,j} \lambda_h^i \lambda_v^j \right) = X(\lambda_h, \lambda_v) - \sum_{j=0}^{\infty} x_{0,j} \lambda_v^j - \sum_{i=0}^{\infty} x_{i,0} \lambda_h^i + x_{0,0}$ , see also [13]. By writing the model (7) in terms of power series we therefore get

$$H(\lambda_h, \lambda_v) X(\lambda_h, \lambda_v) = V(\lambda_h, \lambda_v) U(\lambda_h, \lambda_v) + (E - A_1 \lambda_v) \sum_{j=0}^{\infty} x_{0,j} \lambda_v^j + (E - A_2 \lambda_h) \sum_{i=0}^{\infty} x_{i,0} \lambda_h^i - E x_{0,0} - B_1 \lambda_v \sum_{j=0}^{\infty} u_{0,j} \lambda_v^j - B_2 \lambda_h \sum_{i=0}^{\infty} u_{i,0} \lambda_h^i,$$

On the other hand, if  $x_{i,-1} = x_{-1,j} = 0$  and  $u_{i,-1} = u_{-1,j} = 0$  for  $i, j = 0, 1, 2, \dots$ , according to (7), it can be easily tested

that

$$\begin{aligned} B_2 u_{i-1,0} &= E x_{i,0} - A_2 x_{i-1,0}, \text{ for } i = 1, 2, \dots \\ B_1 u_{0,j-1} &= E x_{0,j} - A_1 x_{0,j-1}, \text{ for } j = 1, 2, \dots \\ x_{0,0} &= 0. \end{aligned} \quad (12)$$

It follows that

$$\begin{aligned} & B_1 \lambda_v \sum_{j=0}^{\infty} u_{0,j} \lambda_v^j + B_2 \lambda_h \sum_{i=0}^{\infty} u_{i,0} \lambda_h^i \\ &= \sum_{j=1}^{\infty} (E x_{0,j} - A_1 x_{0,j-1}) \lambda_v^j + \sum_{i=1}^{\infty} (E x_{i,0} - A_2 x_{i-1,0}) \lambda_h^i \\ &= (E - A_1 \lambda_v) \sum_{j=0}^{\infty} x_{0,j} \lambda_v^j + (E - A_2 \lambda_h) \sum_{i=0}^{\infty} x_{i,0} \lambda_h^i - 2x_{0,0}. \end{aligned} \quad (13)$$

Therefore, (7) expressed in terms of formal power series becomes

$$\begin{aligned} H(\lambda_h, \lambda_v) X(\lambda_h, \lambda_v) &= V(\lambda_h, \lambda_v) U(\lambda_h, \lambda_v), \\ Y(\lambda_h, \lambda_v) &= CX(\lambda_h, \lambda_v) + DU(\lambda_h, \lambda_v), \end{aligned}$$

leading to

$$\begin{aligned} (\det H(\lambda_h, \lambda_v)) Y(\lambda_h, \lambda_v) &= \left( \text{Cadj} H(\lambda_h, \lambda_v) V(\lambda_h, \lambda_v) \right. \\ &\quad \left. + D \det H(\lambda_h, \lambda_v) \right) U(\lambda_h, \lambda_v), \end{aligned}$$

which is clearly equivalent to (6).  $\blacksquare$

By using the complex variables  $z_1$  and  $z_2$  in place of the indeterminates  $\lambda_h$  and  $\lambda_v$  in the association of a signal with a power series representation, we recall that a necessary and sufficient condition for the latent variable model (7) to be asymptotically stable for any set of admissible boundary conditions is that the determinant of  $H(z_1, z_2)$  differ from zero for all  $0 < |z_1| \leq 1$  and  $0 < |z_2| \leq 1$ , see [26, Theorem 3]. In Section II-A it will be proved that every set of boundary conditions (8) is admissible for the latent variable realisation in Theorem 1, so that the condition  $\det H(z_1, z_2) \neq 0$  for all  $0 < |z_1| \leq 1$  and  $0 < |z_2| \leq 1$  guarantees asymptotic stability of (7) for any boundary condition (8).

Now, consider the input-output process (6). Assume that the polynomial  $p(z_1, z_2) = \sum_{i=0}^n \sum_{j=0}^m a_{i,j} z_1^i z_2^j$  has no nonessential singularities of the second kind on the unit bi-torus  $\mathbb{T} := \{(z_1, z_2) \in \mathbb{C} \times \mathbb{C} \mid |z_1| = 1, |z_2| = 1\}^1$ . Then, the input-output process (6) is BIBO stable if and only if  $p(z_1, z_2)$  differs from zero for all  $|z_1| \leq 1$  and  $|z_2| \leq 1$ , [9]. Consider a BIBO stable input-output process (6). If, say,  $n < m$ , the corresponding latent variable model given in Theorem 1 is such that  $\det H(z_1, z_2) = z_1^{n(m-n)} p(z_1, z_2)$ . For all  $0 < |z_1| \leq 1$  the complex number  $z_1^{n(m-n)}$  differs from zero, while  $p(z_1, z_2)$  differs from zero for all  $|z_1| \leq 1$  and  $|z_2| \leq 1$ . As a result, the polynomial  $z_1^{n(m-n)} p(z_1, z_2)$  differs from zero for all  $0 < |z_1| \leq 1$  and  $0 < |z_2| \leq 1$ , which is equivalent to saying that  $\det H(z_1, z_2) \neq 0$  for all  $0 < |z_1| \leq 1$  and  $0 < |z_2| \leq 1$ .

<sup>1</sup>When  $p(z_1, z_2)$  has nonessential singularities of the second kind on the unit bi-torus, the input-output process may be BIBO stable, see [9], but any Fornasini-Marchesini realisation will result to be non-asymptotically stable, see [26, Lemma 3].

Hence, in view of Theorem 3 in [26] it follows that the latent variable model (7) is asymptotically stable.

**Example 2:** Consider a simple case where  $(n, m) = (2, 3)$ . By Theorem 1, we have

$$H(\lambda_h, \lambda_v) = \begin{bmatrix} \lambda_h & 0 & 0 & 0 & 0 \\ 0 & \lambda_h & 0 & 0 & 0 \\ -\lambda_h \lambda_v & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & -\lambda_h \lambda_v & 0 & 0 & 1 \\ 0 & 0 & -\lambda_h \lambda_v & -\lambda_h & -\lambda_v \\ a_{2,3} \lambda_h \lambda_v \\ a_{1,3} \lambda_h \lambda_v + a_{0,3} \lambda_v \\ a_{2,2} \lambda_h \lambda_v \\ a_{2,1} \lambda_h \lambda_v + a_{2,0} \lambda_h \\ a_{1,2} \lambda_h \lambda_v + a_{0,2} \lambda_v \\ 1 + a_{1,1} \lambda_h \lambda_v + a_{0,1} \lambda_v + a_{1,0} \lambda_h \end{bmatrix}, \quad (14)$$

$$V(\lambda_h, \lambda_v) = \begin{bmatrix} \bar{b}_{2,3} \lambda_h \lambda_v \\ \bar{b}_{1,3} \lambda_h \lambda_v + \bar{b}_{0,3} \lambda_v \\ \bar{b}_{2,2} \lambda_h \lambda_v \\ \bar{b}_{2,1} \lambda_h \lambda_v + \bar{b}_{2,0} \lambda_h \\ \bar{b}_{1,2} \lambda_h \lambda_v + \bar{b}_{0,2} \lambda_v \\ \bar{b}_{1,1} \lambda_h \lambda_v + \bar{b}_{1,0} \lambda_h + \bar{b}_{0,1} \lambda_v \end{bmatrix}.$$

A simple computation shows that

$$\det(H(\lambda_h, \lambda_v)) = \lambda_h^2 \sum_{i=0}^2 \sum_{j=0}^3 a_{i,j} \lambda_h^i \lambda_v^j,$$

$$\text{Cadj}(H(\lambda_h, \lambda_v)) = \lambda_h^2 [\lambda_h \lambda_v^2 \quad \lambda_v^2 \quad \lambda_h \lambda_v \quad \lambda_h \quad \lambda_v \quad 1].$$

Therefore, it is readily seen that

$$\text{Cadj}(H(\lambda_h, \lambda_v)) V(\lambda_h, \lambda_v) = \lambda_h^2 \sum_{i=0}^2 \sum_{j=0}^2 \bar{b}_{i,j} \lambda_h^i \lambda_v^j.$$

Thus, (14) is a realisation of (6).

#### A. Recursibility

With the boundary conditions (8), it is easy to check that the solution of (7) is unique and can be determined recursively.

For the case where  $n = m$ , the matrix  $E$  in the realisation (7) is the identity matrix. The model thus obtained is therefore a Fornasini-Marchesini model in Kurek form, [15]. As such, the latent variable  $x$  can be computed recursively when the boundary conditions (8) are assigned.

For the case where  $n < m$ , the matrices appearing in the realisation (7) are in the form given in (9). We may then partition the latent variable  $x$  conformably, so as to obtain

$$x'_{i,j+1} = A_0^{22} x'_{i,j} + A_0^{21} x''_{i,j} + A_1^{21} x''_{i+1,j} + B_0^2 u_{i,j} + B_1^2 u_{i+1,j} \quad (15)$$

$$x''_{i+1,j+1} = A_0^{12} x'_{i,j} + A_0^{11} x''_{i+1,j} + A_2^{11} x''_{i,j+1} + B_0^1 u_{i,j} + B_1^1 u_{i+1,j} + B_2^1 u_{i,j+1} \quad (16)$$

$$y_{i,j} = x''_{i,j}. \quad (17)$$

Consider for simplicity that (15-17) are defined over the bounded frame  $[-1, N] \times [-1, M]$ . Let the boundary conditions associated with (15-16) be as in (8), so that, in particular,  $x'$  is zero on  $[-1, N] \times \{-1\}$ , while  $x''$  and  $u$  are zero on  $[-1, N] \times \{-1\} \cup \{-1\} \times [-1, M]$ . Note that  $x'_{0,0}$  can be computed from (15), since  $x'_{0,-1}, x'_{-1,-1}, x''_{0,-1}, u_{-1,-1}$  and  $u_{0,-1}$  are at zero. It clearly follows that  $x'_{0,0} = 0$ . Similarly, since all the values of  $x', x''$  and  $u$  are fixed at zero on  $[-1, N] \times \{-1\}$ , it follows that by using (15) the component  $x'_{i,0}$  can be computed (and its value is zero) for all  $i = 0, 1, \dots, N-1$ . The value  $x'_{N,0}$  does not influence the output over the bounded frame of interest, due to the fact that  $y$  depends on  $x'$  only through  $x''$  in view of (17). We can now compute  $x''$  on the same row. In fact, now  $x''_{i,0}$  can be computed for all  $i = 0, \dots, N$  by using (16), since  $x'_{i-1,-1}, x''_{i-1,-1}, x'_{i-1,0}, u_{i-1,-1}, u_{i-1}, u_{i-1,0}$  are all given. This yields  $y_{i,0}$  for all  $i = 0, \dots, N$ . At this point, one may proceed with the computation of  $x'_{i,1}$  for all  $i = 0, \dots, N-1$  by using (15) again. In fact, to that end the values  $x'_{i,0}, x''_{i,0}, x'_{i+1,0}, u_{i,0}$  and  $u_{i+1,0}$  are required. These values have been computed in the previous steps. Then again, by exploiting (16) one can compute  $x''_{i,1}$  (and then also  $y_{i,1}$ ) for all  $i = 0, \dots, N$ . This procedure for the computation of  $x'_{i,j}$  and  $x''_{i,j}$  for all  $i = 0, \dots, N-1$  and  $i = 0, \dots, N$ , respectively, can be repeated for all  $j = 1, 2, \dots, M$ , so that the output can be recursively computed all over the bounded frame. Notice that, even if the latent variable component  $x'_{i,j+1}$  depends on the input  $u_{i+1,j}$ , this does not affect the possibility of recursively computing the output on the bounded frame of interest given the input since, as aforementioned, the output  $y$  does not depend explicitly on  $x'$ , but indirectly through  $x''$ . Similar considerations can be made for the case of  $n > m$ .

It turns out that in all three cases (i.e.,  $n = m, n < m$  and  $n > m$ ), when the boundary conditions of (7) are defined in a suitable way as in (8), the latent variable update equation of (7) can be iterated recursively in order to find  $x_{i,j}$  for all  $i, j \geq 0$ .

## B. Order reduction

Theorem 1 provides a latent variable realisation of size  $n \cdot m$  for a general 2-D process described by the input-output relation (6). This means that the latent variable  $x$  is an  $n \cdot m$  vector. If some of the coefficients  $a_{i,j}$  and  $\bar{b}_{i,j}$  in (6) are zero, the size of the realisation may be reduced further by removing some components of the latent variable  $x$ . Denote the  $k$ -th element of  $x$  by  $x[k]$ . If  $x[k]$  is removed from  $x$ , then the corresponding  $k$ -th row of  $A_i$  and  $B_i$ , and the  $k$ -th column of  $A_i$  and  $C$ ,  $i = 0, 1, 2$ , are deleted so that a realisation with smaller size is obtained. The following algorithm proposes a method for the reduction of the size of the realisation based on the coefficients of particular models.

*Algorithm 1:* Denote the  $(k, l)$ -th entry of matrix  $H(\lambda_h, \lambda_v)$  by  $H[k, l]$ . Consider the following series of sets denoted by  $\rho_k$  where  $k = 1, 2, 3, \dots, nm - 1$ :

$$\rho_k = \{A_0[k, nm], A_1[k, nm], A_2[k, nm], B_0[k, 1], B_1[k, 1], B_2[k, 1]\} \cup \rho_{s_1} \cup \rho_{s_2} \cup \dots \cup \rho_{s_j} \quad (18)$$

if there exists  $s_i \in \{1, \dots, k-1\}$ ,  $i = 1, 2, \dots, j$  such that

$$A_0[k, s_i] + A_1[k, s_i] + A_2[k, s_i] \neq 0. \quad (19)$$

If  $j = 0$ , i.e. if there does not exist  $s_i$  such that (19) holds, then

$$\rho_k = \{A_0[k, nm], A_1[k, nm], A_2[k, nm], B_0[k, 1], B_1[k, 1], B_2[k, 1]\}. \quad (20)$$

For a certain  $k$ , the  $k$ -th element of  $x$  can be deleted if  $\rho_k = \{0\}$ .

To show the basic idea of this algorithm, consider the realisation (14) for the case  $(n, m) = (2, 3)$ . It can be found that for the special case  $a_{2,3} = 0$  and  $\bar{b}_{2,3} = 0$ ,  $\rho_1 = \{0\}$  because the  $(1, 6)$ -th entries of all matrices are zero, and there does not exist  $s_i \in \{0, \dots, k-1\}$  such that (19) is satisfied because  $k$  is just 1. By Algorithm 1, the first row and the first column of  $H(\lambda_h, \lambda_v)$  can be removed, and the new  $H(\lambda_h, \lambda_v)$  is denoted by  $\hat{H}(\lambda_h, \lambda_v)$ . Then,

$$\det(\hat{H}(\lambda_h, \lambda_v)) = \lambda_h \sum_{i=0}^2 \sum_{j=0}^3 a_{i,j} \lambda_h^i \lambda_v^j, \\ [0 \ \dots \ 0 \ 1] \text{adj}(\hat{H}(\lambda_h, \lambda_v)) = \lambda_h [\lambda_v^2 \ \lambda_h \lambda_v \ \lambda_h \ \lambda_v \ 1].$$

Thus, the realisation still holds. Consider another special case where  $a_{1,3}, a_{0,3}, \bar{b}_{13}, \bar{b}_{0,3}$  are all equal to zero. Also in this case, the deletion of the corresponding rows and columns from  $H(\lambda_h, \lambda_v)$  does not change the input-output relationship.

However, when the coefficients  $a_{2,2}, \bar{b}_{22}$  are zero, the third row and the third column of  $H(\lambda_h, \lambda_v)$  cannot be removed in general because the entry  $H[3, 1]$  may not be zero. Such deletion, which leads to a new  $H(\lambda_h, \lambda_v)$  denoted by  $\tilde{H}(\lambda_h, \lambda_v)$ , would lead to

$$\det(\tilde{H}(\lambda_h, \lambda_v)) = \lambda_h^2 \sum_{i=0}^1 \sum_{j=0}^3 a_{i,j} \lambda_h^i \lambda_v^j + a_{2,0} \lambda_h^2 + a_{2,1} \lambda_h \lambda_v^2.$$

Obviously the term  $a_{2,3} \lambda_h^2 \lambda_v^3$  also disappears from  $\det(\tilde{H}(\lambda_h, \lambda_v))$  with the term  $a_{2,2} \lambda_h^2 \lambda_v^2$ , and  $\bar{b}_{2,3} \lambda_h^2 \lambda_v^2$  would not appear since

$$[0 \ \dots \ 0 \ 1] \text{adj}(\tilde{H}(\lambda_h, \lambda_v)) = \lambda_h^2 [0 \ \lambda_v^2 \ \lambda_h \ \lambda_v \ 1].$$

Therefore, whether the third row and the third column can be deleted not only depends on  $a_{2,2}, \bar{b}_{22}$ , but also on  $a_{2,3}$  and  $\bar{b}_{2,3}$ . Consistently, in Algorithm 1 the set  $\rho_3$  is defined as  $\{a_{2,2}, \bar{b}_{22}\} \cup \rho_1 = \{a_{2,2}, \bar{b}_{22}, a_{2,3}, \bar{b}_{2,3}\}$ , and all the coefficients in  $\rho_3$  must be zero to allow the deletion of the third component of the latent variable  $x$ .

Stated differently, for a particular  $k$ , even if  $H[k, nm] = 0$  and correspondingly  $V[k] = 0$ , we need to check if  $H[k, s] = 0$  for  $s = 1, \dots, k-1$ . Notice that we do not care about  $H[k, s]$  for  $s = k, \dots, nm-1$ , because these values are all zeros according to Theorem 1.

To further verify and illustrate the proposed algorithm, consider the following example.

**Example 3:** Consider a 2-D system ruled by

$$\begin{aligned} & \left(1 + a_{0,1}\lambda_v + a_{1,0}\lambda_h + a_{1,1}\lambda_h\lambda_v + a_{2,4}\lambda_h^2\lambda_v^4\right)Y(\lambda_h, \lambda_v) \\ & = \left(b_{0,1}\lambda_v + b_{1,0}\lambda_h + b_{1,1}\lambda_h\lambda_v + b_{2,4}\lambda_h^2\lambda_v^4\right)U(\lambda_h, \lambda_v). \end{aligned} \quad (21)$$

By Theorem 1, a realisation in the form of (7) of order 8 can be obtained. For the general (2,4)-th order system, according to Algorithm 1, we have

$$\begin{aligned} \rho_1 &= \{a_{24}, b_{24}\} \\ \rho_2 &= \{a_{04}, a_{14}, b_{04}, b_{14}\} \\ \rho_3 &= \{a_{23}, b_{23}\} \cup \rho_1 \\ \rho_4 &= \{a_{03}, a_{13}, b_{03}, b_{13}\} \cup \rho_2 \\ \rho_5 &= \{a_{22}, b_{22}\} \cup \rho_3 \\ \rho_6 &= \{a_{20}, a_{21}, b_{20}, b_{21}\} \\ \rho_7 &= \{a_{02}, a_{12}, b_{01}, b_{12}\} \cup \rho_4. \end{aligned}$$

For the particular system (21), it is found that all elements in  $\rho_2, \rho_4, \rho_6$  and  $\rho_7$  are zero. As such, the 2nd, 4th, 6th and 7th components of the latent variable  $x$  can be removed. As a result of this, the realisation becomes

$$\begin{aligned} E &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad A_0 = \begin{bmatrix} 0 & 0 & 0 & -a_{2,4} \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -a_{1,1} \end{bmatrix}, \\ A_1 &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -a_{0,1} \end{bmatrix}, \quad A_2 = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -a_{1,0} \end{bmatrix}, \\ B_0 &= \begin{bmatrix} b_{2,4} \\ 0 \\ 0 \\ b_{1,1} \end{bmatrix}, \quad B_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ b_{0,1} \end{bmatrix}, \quad B_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ b_{1,0} \end{bmatrix}, \quad C = [0 \quad 0 \quad 0 \quad 1]. \end{aligned}$$

### III. CONCLUSIONS

This paper investigates the possibility of representing a causal 2-D linear input-output process by means of an implicit latent variable model. The advantage of the proposed model is that it provides system matrices of much smaller size with respect to other realisations presented so far in the literature. Indeed, for the general  $(n, m)$ -th order 2-D system in the form of (6), the size of the realisation by using the proposed model is  $n \cdot m$ . Importantly, this goal is achieved without affecting recursibility of the latent variable model thus obtained. For the Fornasini-Marchesini second model, on the other hand, the size of the realisation given in [24] is  $nm + m + n$ , which is the smallest dimension of all the realisations available in literature using Fornasini-Marchesini realisations [3], [6].

Another contribution of this paper is an algorithm to further reduce the size of the realisation when some of the coefficients of the transfer function are zero.

### REFERENCES

- [1] N. Amann, D.H. Owens and E. Rogers, 2D systems theory applied to learning control systems, In *Proceedings of the 33rd IEEE Conference on Decision and Control*, vol. 2, Lake Buena Vista, FL, USA, pp. 985–986, 1994.
- [2] G.E. Antoniou, P.N. Paraskevopoulos and S.J. Varoufakis, Minimal state-space realization of factorable 2-D transfer functions, *IEEE Transactions on Circuits and Systems*, vol. 35, no. 8, pp. 1055–1058, 1988.
- [3] M. Bisiacco, E. Fornasini and G. Marchesini, Dynamic regulation of 2D systems: a state-space approach, *Linear Algebra and Its Applications*, vol. 122/123/124, pp. 195–218, 1989.
- [4] N.K. Bose. *Multidimensional Systems Theory and Applications*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 2003.
- [5] C. Du, and L. Xie, *H<sub>∞</sub> Control and Filtering of Two-Dimensional Systems*, Springer, 2002.
- [6] E. Fornasini and G. Marchesini, State-space realization theory of two-dimensional filters, *IEEE Transactions on Automatic Control*, vol. AC-21, no. 4, pp. 484–492, 1976.
- [7] E. Fornasini and G. Marchesini, Doubly-indexed dynamical systems: state-space models and structural properties,” *Mathematical Systems Theory*, vol. 12, pp. 59–72, 1978.
- [8] K. Galkowski, *State-space realizations of linear 2-D systems with extensions to the general nD case*, Lecture Notes in Control and Information Sciences, Springer, 2001.
- [9] D. Goodman, Some stability properties of two-dimensional linear shift-invariant digital filters, *IEEE Transactions on Circuits and Systems*, vol. CAS-24, no. 4, pp. 201–208, 1977.
- [10] T. Hinamoto, and A. Doi, and M. Muneyasu. 2-D adaptive state-space filters based on the Fornasini-Marchesini second model, *IEEE Transactions on Circuits and Systems - II*, Vol. 44, pp. 667–670, 1997.
- [11] T. Kaczorek, *Two-Dimensional Linear Systems*, Springer-Verlag, 1985.
- [12] T. Kaczorek, The singular general model of 2-D systems and its solution, *IEEE Transactions on Automatic Control*, vol. 33, no. 11, pp.1060–1061, 1988.
- [13] T. Kaczorek, General response formula and minimum energy control for the general singular model of 2-D systems, *IEEE Transactions on Automatic Control*, vol. 35, pp. 433–436, 1990.
- [14] T. Kaczorek, *Positive 1D and 2D systems*, Springer, 2002.
- [15] J.E. Kurek, The general state-space model for a two-dimensional linear digital systems, *IEEE Transactions on Automatic Control*, vol. AC-30, no. 6, pp. 600–602, 1985.
- [16] J.E. Kurek, and M.B. Zaremba, Iterative learning control synthesis based on 2-D system theory, *IEEE Transactions on Automatic Control*, vol. 38, no. 1, pp. 121–125, 1993.
- [17] F.L. Lewis and B.G. Mertzios, On the Analysis of Two-Dimensional Discrete Singular Systems, *Circuits Systems and Signal Processing*, vol. 11, no. 3, pp. 399–419, 1992.
- [18] C. Li and M.S. Fadali, Optimal control of 2-D systems, *IEEE Transactions on Automatic Control*, vol. 36, no. 2, pp. 223–228, 1991.
- [19] W.S. Lu, W.S. and E.B. Lee, Stability analysis for two-dimensional systems via a Lyapunov approach, *IEEE Transactions on Circuits and Systems*, vol. 32, pp. 61–68, 1985.
- [20] D. Luenberger, Dynamic equations in descriptor form, *IEEE Transactions on Automatic Control*, vol. AC-22, no. 3, pp. 310–319, 1977.
- [21] D. Luenberger, Time-invariant descriptor systems, *Automatica*, vol. 14, no. 5, pp. 473–480, 1978.
- [22] L. Ntogramatzidis, R. Yang, and M. Cantoni, “On the realisation of 2-D linear systems with second-order Fornasini-Marchesini models,” Technical Report, TR06-001, Department of Electrical and Electronic Engineering, The University of Melbourne, 1/6/2005.
- [23] R. Roesser, A discrete state-space model for linear image processing, *IEEE Transactions on Automatic Control*, vol. AC-20, no. 1, pp. 1–10, 1975.
- [24] L. Xu, L. Wu, Q. Wu, Z. Lin and Y. Xiao, On the realization of 2D discrete systems by Fornasini-Marchesini model, *International Journal of Control, Automation, and Systems*, vol. 3, pp. 631–639, 2005.
- [25] J.Y. Zhang, and W. Steenaart, A very fast Kalman filter for image restoration, *IEEE International Symposium on Circuits and Systems*, vol. 1, pp. 250–253, May, 1990.
- [26] Yun Zou, and S.L. Campbell, The jump behavior and stability analysis for 2-D singular systems, *Multidimensional Systems and Signal Processing*, vol. 11, no. 4, pp. 321–338, 2000.