State estimation and fault detection for 2-D discrete-time systems

Dalil Ichalal [1], Benoît Marx[2] and José Ragot[2]

Abstract— **This paper presents an observer based sensor fault diagnosis for 2-D systems. Firstly, a state observer is designed by taking into account the measurements in a fixed length sliding window. Secondly, structured residual signals are generated by a bench of finite memory state observers which allow to detect and isolate sensor faults affecting the system. Finally, simulation results are given to illustrate the proposed approach.**

I. INTRODUCTION

Since thirty years, a growing interest for systems whose state depends on several variables can be noted. Even if most of the published works concern two dimensional (2-D) systems, some concepts can also be addressed for general multidimensional systems [2] [3]. In image processing, 2-D filters are well known and intensively applied. The main fields of automatic control have also been revisited through 2-D systems: modeling, stability analysis, control and state estimation. More recently, several works have been proposed in the framework of fault diagnosis, for sensor fault detection and unknown input estimation.

Among the precursor works, [10] [11] have introduced the first formulation of double-indexed systems resulting in particular in the 2-D digital filters [1]. This allowed to develop the concept of 2-D transfer function [23] [8] [22], to propose its calculation from a state equation [30] [15] [14] and to analyze its observability and stability properties [17] [21] [19]. Many publications were devoted to the control of 2-D systems in various fields: iterative learning control [29], quadratic control [24], H_∞ control [20], trajectory tracking [31]. State estimation has also been addressed: the Luenberger-like observer design has been extended to 2-D systems in [6] [13] [32]. Then, these 2-D observers have been used for state and unknown input estimation and fault diagnosis [5] [4] [7] . Amon the 2-D model literature, a lot of works concern the methodological aspects, however only a few of them are dedicated to application purposes. Nevertheless, let us cite [9] and [12] in the metallurgical field, [16] for an application related to the vehicle traffic and [28] in image processing.

In what follows, our attention is focused on the fault diagnosis by extending the use of the parity space developed in the context of 1-D systems [27] to the case of 2-D systems. The faults to be detected and isolated affect the sensors, but other types of faults could be taken into account, in particular those affecting the actuators of the system.

II. 2-D SYSTEM MODELING

In the following the Fornasini-Marchesini [11] model is used. Its structure is given by:

$$\begin{cases} x(i{+}1,j{+}1) = A_1x(i{+}1,j) + A_2x(i,j{+}1) + Bu(i,j) \\ y(i,j) = Cx(i,j) \end{cases} \tag{1}$$

where $x(i,j) \in \mathcal{R}^{n_x}$, $u(i,j) \in \mathcal{R}^{n_u}$ and $y(i,j) \in \mathcal{R}^{n_y}$ represent respectively the state, input and output vectors. A_1, A_2, B, C, D are real matrices with appropriate dimensions. The system is also expressed in a conventional form by (A_1, A_2, B, C) and its transfer function is defined by:

$$W(z_1, z_2) = C(I - A_1z_1 - A_2z_2)^{-1}Bz_1z_2 \tag{2}$$

The expression (2) can be used for stability analysis [16] and, more generally, to highlight its structural properties [18]. The figure 1 depicts the simulation scheme of the system (1). This simulation requires the knowledge of the initial boundary conditions on the state in the directions i and j and the sequence of calculation order.

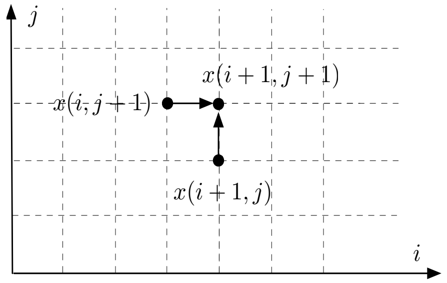

Fig. 1. State calculus scheme

Remark 1: In many situations, especially when one of the directions, j for instance, indexes a spatial direction, the evolution of the system is confined in a domain of the type $\mathcal{D}_N = (i,j) : i \geq 0, 0 \leq j \leq N$ [31].
The following notations are used:

$$\begin{cases} A^{(i,j)} &= A_1A^{(i-1,j)} + A_2A^{(i,j-1)} \\ A^{(0,0)} &= I \\ A^{(i,j)} &= 0 \quad if\ i < 0 \quad or \quad if\ j < 0 \end{cases} \tag{3}$$

Some useful definitions are given below.

Definition 1: The model is observable if each couple $x(h+1,k), x(h,k+1)$ can be computed from the future values of the inputs and the outputs $y(i,j), u(i,j), i \geq h$ and $j \geq k$.

[1] Dalil Ichalal is with IBISC-Lab, Evry Val d'Essonne University, 40, rue de Pelvoux, 91020, Evry Courcouronnes, France (email: dalil.ichalal@univ-evry.fr)

[2] Benoît Marx and José Ragot are with Université de Lorraine, CNRS, CRAN, F-54000 Nancy, France (email: benoit.marx@univ-lorraine.fr, jose.ragot@univ-lorraine.fr)

Definition 2: The model is reconstructible if each couple $x(h+1,k), x(h,k+1)$ can be computed from the past values of the inputs and the outputs $y(i,j), u(i,j), i \leq h$ et $j \leq k$.

Lemma 1: [17] The model is reconstructible if the pair $(C, A_1 r + A_2 s)$ is reconstructible for all $r, s \in \mathcal{C}$.

III. Finite memory observer

The finite memory observer aims to reconstruct the system state by using only the input and output measurements made on a finite time horizon. Let us consider the discrete-time 2-D (1). Firstly, the system state $x(i,j)$ is expressed with respect to the initial conditions and the inputs, then it is written on a finite horizon.

A. State and output expressions with respect to inputs and state initial conditions

Theorem 1: The state of the system (1) is expressed with respect to its initial conditions and inputs as follows:

$$\begin{aligned} x(i,j) =& \sum_{k=1}^{j} A^{(j-k,i-1)} A_2 x(0,k) + \sum_{k=1}^{i} A^{(j-1,i-k)} A_1 x(k,0) \\ &+ \sum_{n=0}^{j-1} \sum_{m=0}^{i-1} A^{(j-n-1,i-m-1)} Bu(m,n) \end{aligned} \tag{4}$$

where the matrices $A^{(i,j)}$ are defined in (3). Consequently, the output of the system is expressed as:

$$\begin{aligned} y(i,j) =& \sum_{k=1}^{j} CA^{(j-k,i-1)} A_2 x(0,k) + \sum_{k=1}^{i} CA^{(j-1,i-k)} A_1 x(k,0) \\ &+ \sum_{n=0}^{j-1} \sum_{m=0}^{i-1} CA^{(j-n-1,i-m-1)} Bu(m,n) \end{aligned} \tag{5}$$

The figure 2 depicts, for a simple case: $i = 4$ and $j = 3$, both the input and state spaces. The stars stand for the initial state position.

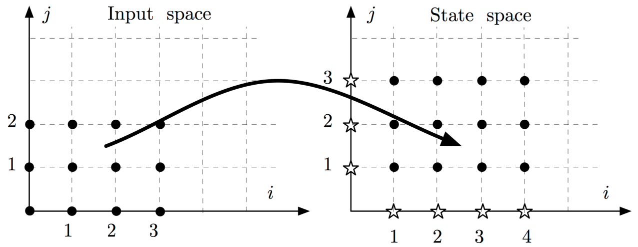

Fig. 2. State computation, from the input space (left) to the state space (right)

Proof: The proof is made by induction. For $(i,j) = (1,1)$, from (1), we have:

$$x(1,1) = A_1 x(1,0) + A_2 x(0,1) + Bu(0,0) \tag{6}$$

which is also given by (4). Now, assuming that (4) is true for (i,j), $x(i+1,j+1)$ is computed. Firstly, from (1) we have:

$$x(i+1,j) = A_1 x(i+1,j-1) + A_2 x(i,j) + Bu(i,j-1) \tag{7}$$

Using (4), equation (7) can be written in terms of the inputs and of the initial states:

$$\begin{aligned} x(i+1,j) =& A_1 \big(\sum_{n=0}^{j-2} \sum_{m=0}^{i} A^{(j-n-2,i-m)} Bu(m,n) + \\ & \sum_{k=1}^{j} A^{(j-1-k,i)} A_2 x(0,k) + \sum_{k=1}^{i+1} A^{(j-2,i+1-k)} A_1 x(k,0) \big) \\ & + Bu(i,j-1) + A_2 \big(\sum_{n=0}^{j-1} \sum_{m=0}^{i-1} A^{(j-n-1,i-m-1)} Bu(m,n) + \\ & \sum_{k=1}^{j} A^{(j-1-k,i-1)} A_2 x(0,k) + \sum_{k=1}^{i} A^{(j-1,i-k)} A_1 x(k,0) \big) \end{aligned} \tag{8}$$

In (8), the term multiplying $x(0,k)$, denoted by p_1, can be simplified as follows:

$$\begin{aligned} p_1 =& A_1 \left(\sum_{k=1}^{j-1} A^{(j-1-k,i)} A_2 x(0,k) \right) \\ & + A_2 \left(\sum_{k=1}^{j} A^{(j-k,i-1)} A_2 x(0,k) \right) \\ =& \sum_{k=0}^{j-1} (A_1 A^{(j-1-k,i)} + A_2 A^{(j-k,i-1)}) A_2 x(0,k) \\ & + A_2^2 A_2 x(0,j) \\ =& \sum_{k=0}^{j} A^{(j-k,i)} A_2 x(0,k) \end{aligned} \tag{9}$$

The term multiplying $x(k,0)$ in (8), denoted p_2, becomes:

$$\begin{aligned} p_2 =& A_1 \sum_{k=1}^{i+1} A^{(j-2,i+1-k)} A_1 x(k,0) \\ & + A_2 \sum_{k=1}^{i} A^{(j-1,i-k)} A_1 x(k,0) \\ =& \sum_{k=1}^{i+1} A^{(j-2,i+1-k)} A_1 x(k,0) \end{aligned} \tag{10}$$

Finally, the term multiplying u in (8), denoted p_3 becomes:

$$\begin{aligned} p_3 =& A_1 \sum_{n=0}^{j-2} \sum_{m=0}^{i} A^{(j-n-2,i-m)} Bu(m,n) \\ & + A_2 \sum_{n=0}^{j-1} \sum_{m=0}^{i-1} A^{(j-n-1,i-m-1)} Bu(m,n) \\ & + Bu(i,j-1) \\ =& \sum_{m=0}^{i} \sum_{n=0}^{j-1} A^{(j-n-1,i-m)} Bu(m,n) \end{aligned} \tag{11}$$

According to (9), (10) and (11), the state (8) becomes:

$$\begin{aligned} x(i+1,j) =& \sum_{k=0}^{j} A^{(j-k,i)} A_2 x(0,k) \\ & + \sum_{k=1}^{i+1} A^{(j-2,i+1-k)} A_1 x(k,0) \\ & + \sum_{m=0}^{i} \sum_{n=0}^{j-1} A^{(j-n-1,i-m)} Bu(m,n) \end{aligned} \tag{12}$$

Then, $x(i, k+1)$ can be obtained analogously and, finally, the two expressions of $x(i+1, j)$ and $x(i, k+1)$ imply (4). The equation (5) is a direct consequence of (4). ■

B. State and output expressions on a finite horizon

The state measurements can now be expressed from the state equation (4) according to the inputs and the initial values of the system state. Without loss of generality, this expression will be established on the window $(i, j) \in [1\!:\!I] \times [1\!:\!J]$ for which the initial states are $x(0, j), j = 1, \dots, J$ and $x(i, 0), i = 1, \dots, I$.

By stating the equations in the interval $(i, j) \in [1\!:\!I] \times [1\!:\!J]$, one can establish the following equation expressing the state and output vectors x and y gathering all the states and the outputs in the considered window with respect to the initial conditions x_0 and the inputs u in the same window:

$$x = F_x x_0 + F_u u \tag{13a}$$

$$y = G_x x_0 + G_u u \tag{13b}$$

with $x_0 \in \mathcal{R}^{(I+J).n_x}$, $u \in \mathcal{R}^{I.J.n_u}$, $x \in \mathcal{R}^{I.J.n_x}$, $y \in \mathcal{R}^{I.(J+2).n_y}$ and with the definitions:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_I \end{bmatrix} \quad x_i = \begin{bmatrix} x(i,1) \\ x(i,2) \\ \vdots \\ x(i,J) \end{bmatrix} \quad x_0 = \begin{bmatrix} x_{.,0} \\ x_{0,.} \end{bmatrix}$$

$$x_{.,0} = \begin{bmatrix} x(1,0) \\ x(2,0) \\ \vdots \\ x(I,0) \end{bmatrix} \quad x_{0,.} = \begin{bmatrix} x(0,1) \\ x(0,2) \\ \vdots \\ x(0,J) \end{bmatrix} \quad u = \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{I-1} \end{bmatrix}$$

$$u_i = \begin{bmatrix} u(i,0) \\ u(i,1) \\ \vdots \\ u(i,J-1) \end{bmatrix} \quad y = \begin{bmatrix} y_{.,0} \\ y_{0,.} \\ y_1 \\ y_2 \\ \vdots \\ y_I \end{bmatrix} \quad y_i = \begin{bmatrix} y(i,1) \\ y(i,2) \\ \vdots \\ y(i,J) \end{bmatrix} \tag{14}$$

where $x_i \in \mathcal{R}^{J.n_x}$, $u_i \in \mathcal{R}^{J.n_u}$. F_x, F_u, G_x and G_u have appropriate dimensions and depend on $A^{(i,j)}$, B and C (details on these matrices will be given in an example).

Remark 2: The equation (13) clearly states the influence of the initial state x_0 on the current system state. Moreover, (13) can be used for any window length, provided the matrices F_x, F_u, G_x and G_u are adapted accordingly. However, for a sliding observation window of fixed length $[0 : I] \times [0 : J]$, the matrices F_x, F_u, G_x and G_u are constant and only the definitions (14) are reevaluated.
This remark will naturally be used, by adapting the observation window, when performing the state and output estimations for fault diagnosis.

C. Finite memory observer (FMO) design

Based on (13b) and on the input/output measurements on the observation window $[0 : I] \times [0 : J]$, the initial system state is estimated by minimizing the following criteria:

$$\Phi = \parallel y - G_x x_0 - G_u u \parallel_W^2 \tag{15}$$

where the weighting matrix W allows to privilege the effect of certain observations. If the matrix G_x is full column rank, the estimated initial state is computed from:

$$\hat{x}_0 = (G_x^T W G_x)^{-1} G_x^T W (y - G_u u) \tag{16}$$

Based on (13) and (16), the following FMO can be proposed

$$\begin{aligned} \hat{x} = &\quad F_x \hat{x}_0 + F_u u \\ \hat{y} = &\quad G_x \hat{x}_0 + G_u u \end{aligned} \tag{17}$$

The state estimation on the window $[0 : I] \times [0 : J]$ is applicable for all other observation sequences of fixed length, especially, for the window $[1 : I+1] \times [0 : J]$, the variable i being considered as the discrete time and the variable j indexing a constant number of states. As pointed out in the remark 2, the structure of (16) and (17) are valid for a sliding window. Only the values of u and y must be updated, while the matrices preserve their structures and constant parameters.

IV. Sensor fault detection

A. Residual generation for fault detection

If a sensor fault affects the system, (1) becomes

$$\begin{aligned} & x(i+1, j+1) = A_1 x(i+1, j) + A_2 x(i, j+1) + Bu(i, j) \\ & y(i, j) = Cx(i, j) + f \end{aligned} \tag{18}$$

where $f = [f_1 \dots f_{n_y}]^T$ is the sensor fault. The output estimation of the FMO (17) allows to generate the output reconstruction error: $\tilde{y}(i, j) = y(i, j) - \hat{y}(i, j)$ that is a sensor fault indicator or *residual*. Its analysis allows to performs sensor fault detection and isolation [26].

B. Example for horizon of length 2

The fault free output vector can be expressed as (19) to highlight the relation between the inputs in $[0 : 1] \times [0 : 1]$ and the outputs in $[0 : 2] \times [0 : 2]$,

$$\begin{bmatrix} y(1,0) \\ y(2,0) \\ \hline y(0,1) \\ y(0,2) \\ \hline y(1,1) \\ y(1,2) \\ \hline y(2,1) \\ y(2,2) \end{bmatrix} = \left[\begin{array}{cc|cc} C & 0 & 0 & 0 \\ 0 & C & 0 & 0 \\ \hline 0 & 0 & C & 0 \\ 0 & 0 & 0 & C \\ \hline CA_1 & 0 & CA_2 & 0 \\ CA_1^2 & 0 & CA_1A_2 & CA_2 \\ \hline CA_2A_1 & CA_1 & CA_2^2 & 0 \\ CA^{(1,1)}A_1 & CA_1^2 & CA^{(1,1)}A_2 & CA_2^2 \end{array} \right] \times \begin{bmatrix} x(1,0) \\ x(2,0) \\ \hline x(0,1) \\ x(0,2) \end{bmatrix} +$$

$$\left[\begin{array}{cc|cc} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \hline CB & 0 & 0 & 0 \\ CA_1B & CB & 0 & 0 \\ \hline CA_2B & 0 & CB & 0 \\ CA^{(1,1)}B & CA_2B & CA_1B & CB \end{array} \right] \begin{bmatrix} u(0,0) \\ u(0,1) \\ \hline u(1,0) \\ u(1,1) \end{bmatrix} \tag{19}$$

In (19) structure of (13) is preserved with G_x and G_u defined by:

$$G_x = \begin{bmatrix} C & 0 & 0 & 0 \\ 0 & C & 0 & 0 \\ 0 & 0 & C & 0 \\ 0 & 0 & 0 & C \\ CA_1 & 0 & CA_2 & 0 \\ CA_1^2 & 0 & CA_1A_2 & CA_2 \\ CA_2A_1 & CA_1 & CA_2^2 & 0 \\ CA^{(1,1)}A_1 & CA_1^2 & CA^{(1,1)}A_2 & CA_2^2 \end{bmatrix}$$
$$G_u = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ CB & 0 & 0 & 0 \\ CA_1B & CB & 0 & 0 \\ CA_2B & 0 & CB & 0 \\ CA^{(1,1)}B & CA_2B & CA_1B & CB \end{bmatrix} \quad (20)$$

The particular structure of G_x can be used to study its rank, that is useful to analyse the rank of $G_x^TWG_x$ in (16).

C. Numerical application

The following example will be processed according to different sensor configurations in order to highlight the sensor fault detection and isolation. The system under consideration is defined by the state matrices:

$$A_1 = \begin{bmatrix} 0.8 & -0.2 & 0 \\ 0 & 0.6 & 0.2 \\ 0.1 & 0 & -0.4 \end{bmatrix} \quad A_2 = \begin{bmatrix} 0.4 & 0.2 & 0.1 \\ -0.1 & 0.5 & -0.3 \\ 0.1 & 0 & 0.15 \end{bmatrix}$$
$$B = \begin{bmatrix} 0.5 \\ -0.5 \\ 0.2 \end{bmatrix} \quad (21)$$

The input control is given by: $u = \sin(t/15\cos(t/17))/2 + 0.1$ The states $x(i,j) \in \mathcal{R}^3$ are defined in the domain $[0 : I] \times [0 : J]$ where I and J will be defined later.

In this example, fault detection and isolation is performed by designing a bank of FMOs, where each observer uses specific measurements and then is sensitive to the faults affecting these outputs while being insensitive to the others.

The states x_1, x_2 and x_3 are computed in the domain $(i,j) \in [0 : 100] \times [1 : 5]$. In order to highlight the fault isolation performance, let us assume that all the states are measured, i.e. $C = I_3$. A sensor fault affect the second output in the interval $(i,j) \in [33 : 43] \times [3 : 4]$. In this example, the FMO using the outputs $\{y_1, y_2, y_3\}$, $\{y_1, y_2\}$, $\{y_1, y_3\}$ and $\{y_2, y_3\}$ are designed.

The figure 3 illustrates the real system states. The figures 4, 5 and 6 depict the estimated states obtained from the FMOs using respectively the outputs $\{y_1, y_2\}$, $\{y_1, y_3\}$ and $\{y_2, y_3\}$ respectively (for lack of space the estimates obtained with the outouts $\{y_1, y_2, y_3\}$) are not presented). The analysis of these figures shows that, even in the presence of the fault, the states are perfectly estimated, provided the output y_2 is not used. It can be clearly seen when computing the output error $\tilde{y}(i,j) = y(i,j) - C\hat{x}(i,j)$.

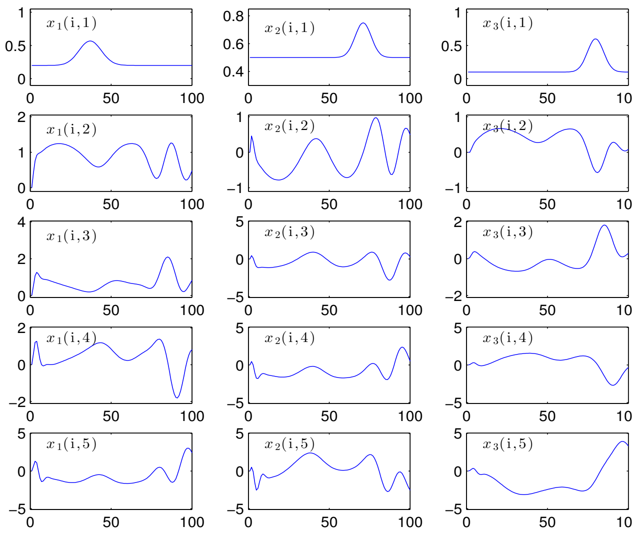

Fig. 3. Real system states

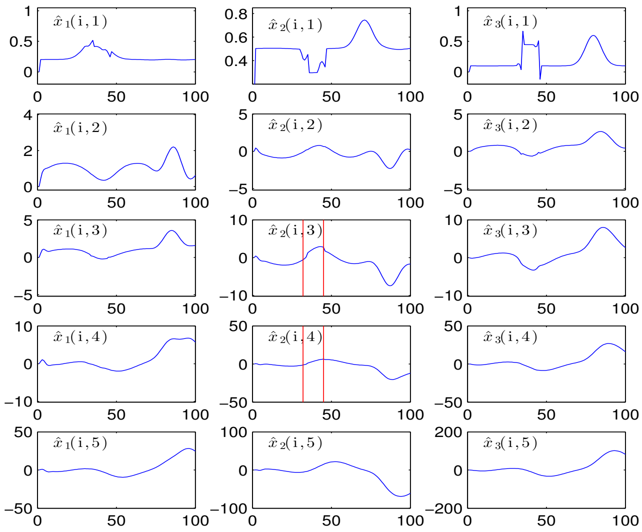

Fig. 4. Estimated state of the FMO using $\{y_1, y_2\}$

The analysis of the graphical results should be linked to fault signatures given in the table I. In this table, the sensitivity of each residual to each possible fault f_i is indicated, providing the theoretical signatures of each fault. Note that these signatures are distinct from each other and thus allows to perform fault isolation.

From a practical point of view, the fault detection and isolation directly results from a comparison, at every sample

	f_1	f_2	f_3
$\tilde{y}_{\{y_1,y_2,y_3\}}$	×	×	×
$\tilde{y}_{\{y_1,y_2\}}$	×	×	.
$\tilde{y}_{\{y_1,y_3\}}$	×	.	×
$\tilde{y}_{\{y_2,y_3\}}$	.	×	×

TABLE I
THEORETICAL FAULT SIGNATURE

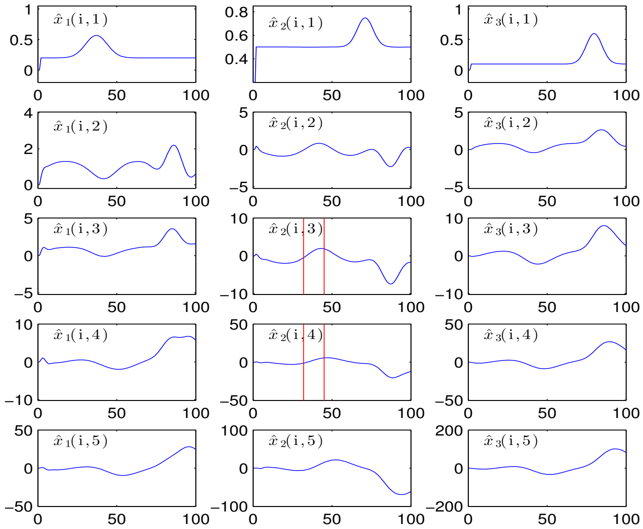

Fig. 5. Estimated state of the FMO using $\{y_1, y_3\}$

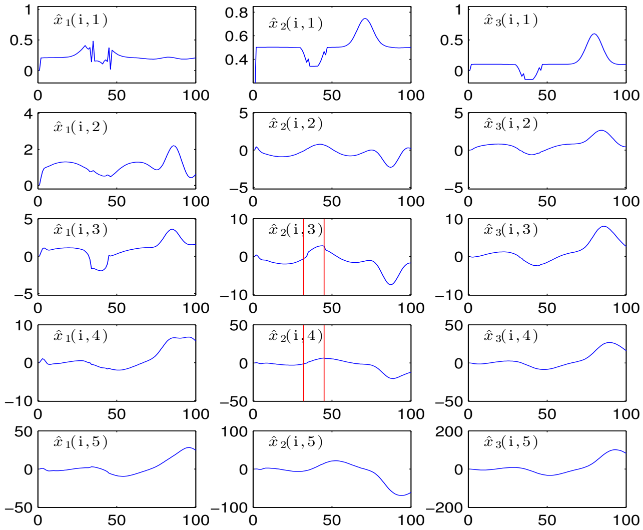

Fig. 6. Estimated state of the FMO using $\{y_2, y_3\}$

time, of the theoretical and experimental fault signatures. The theoretical signatures are given in the table I. The experimental ones are obtained by thresholding the absolute value of the residual signals given by the output reconstruction error of the different FMO. These thresholds are previously chosen according to the amplitude of the faults to be detected and eventually adjusted by learning.

The analysis of the figures 7, 8, 9 and 10 clearly reveals the presence of the faults. The residual signals of the figure 7, provided by the observer using all the outputs (including the faulty ones) are all sensitive to the fault with different degrees of sensitivity. The fault isolation is illustrated in the figures 8, 9 and 10. It is clear that only the observer using $\{y_1, y_3\}$ allows the isolation by comparing the experimental signature of the residual signals in the interval $i \in [33:43]$ with the theoretical signatures of the table I.

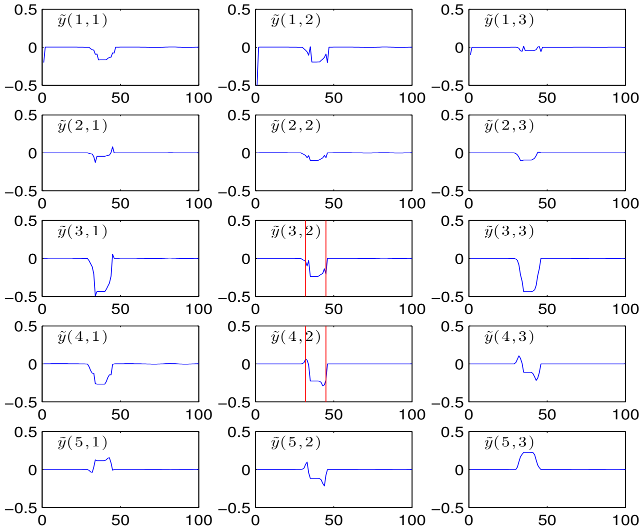

Fig. 7. Residual signal based on $\{y_1, y_2, y_3\}$

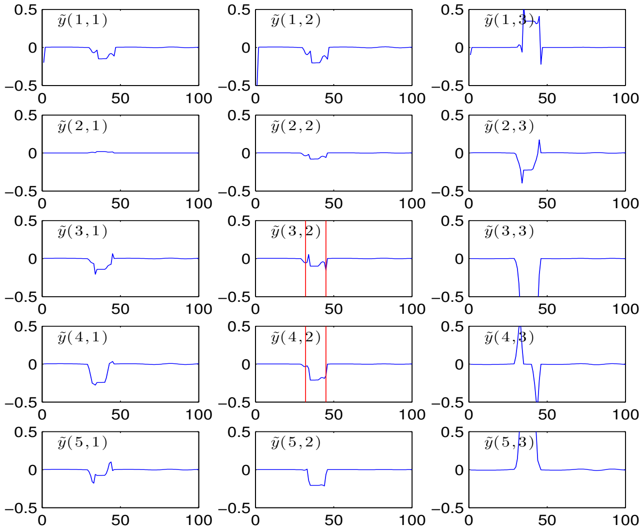

Fig. 8. Residual signal based on $\{y_1, y_2\}$

V. Conclusion

This paper shows how to use a finite memory observer for 2-D systems for the purpose of sensor fault diagnosis. Emphasis is placed on the design of bank of observers in order to be able to isolate faults affecting the plant sensors. Naturally, a fairly straightforward extension of this work would be to extend the obtained results to the detection and isolation of faults affecting the inputs of the system (namely actuator FDI). The detection and isolation of faults affecting the system itself (component faults, which can result in the variations of some system parameters) is in general more tedious.

References

[1] J.F. Abramatic, F. Germain, E. Rosencher. Design of Separable Denominator 2-D Recursive Filters. *IEEE International Conference on Acoustics, Speech and Signal Processing*, Washington DC, USA,1979.

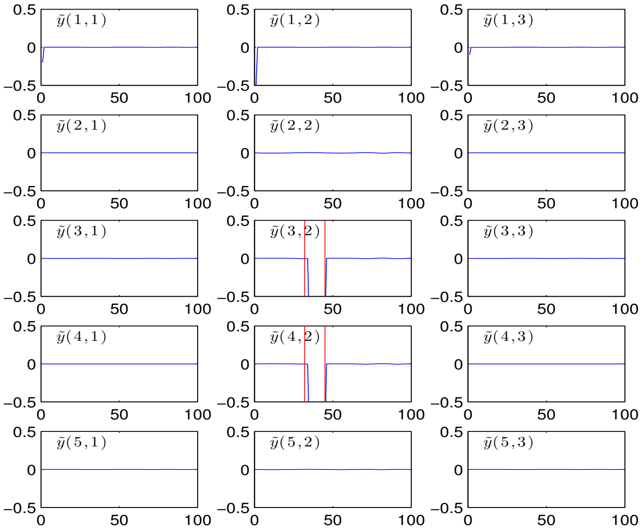

Fig. 9. Residual signal based on $\{y_1, y_3\}$

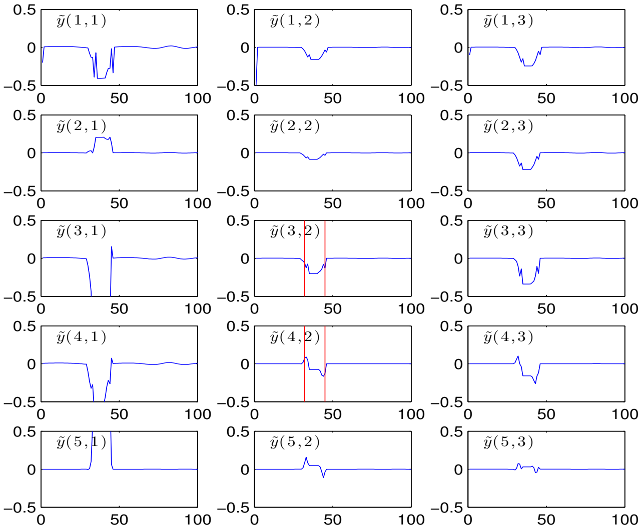

Fig. 10. Residual signal based on $\{y_2, y_3\}$

[2] G.E. Antoniou. Transfer function computation for generalized n-dimensional systems. *Journal of Franklin Institute*, 338 (1), 83-90, 2001.

[3] G.E. Antoniou, M. T. Michael. N-dimensional k-order systems: computing the transfer function. *International Symposium on Signals, Circuits and Systems*, Iasi, Romania, 2005.

[4] A. Baniamerian, N. Meskin, K. Khorasani. Fault Detection and Isolation of Fornasini-Marchesini 2-D Systems: A Geometric Approach. *American Control Conference*, Portland, Oregon, USA, 2014.

[5] M. Bisiacco, M.E. Valcher. Observer-based fault detection and isolation for 2-D state-space models. *Multidimensional Systems and Signal Processing*, 17 (2-3), 219-242, 2006.

[6] M. Bisiacco, M.E. Valcher. Generalized Luenberger observers for 2-D state-space models. *International Workshop on Multidimensional (nD) Systems*, Aveiro, Portugal, 2007.

[7] L. Cao, C. Liu, Y. Wang. Asymptotically stable observer for two-dimensional systems with multiple-channel faults. *Chinese Control Conference*, Dalian, China, 2017.

[8] A. Eydgahi, H. Singh. Computing 2-D transfer function matrix from state-space equations. *IEEE Transactions on Automatic Control*, 31 (4), 259-360, 1986.

[9] S. Foda, P. Agathoklis. Control of the metal rolling process: a multidimensional system approach. *Journal of the Franklin Institute*, 329 (2), 317-332, 1992.

[10] E. Fornasini, G. Marchesini. Algebraic realization theory of two-dimensional filters. *Seminar on "Variable Structure Systems"*, Portland, Oregon, A. Ruberti and R. Mohler eds., Lectures Notes in Economics and Mathematical Systems, 111, 64-82, 1974.

[11] E. Fornasini, G. Marchesini. Doubly indexed dynamical systems: State-space models and structural properties. *Mathematical System Theory*, 12, 59-72, 1978.

[12] K. Galkowski, W. Paszke, E. Rogers, D.H. Owens. Stabilization and robust control of metal rolling modeled as a 2-D linear system. *IFAC World Congress*, Barcelona, Spain, 2002.

[13] G. Izuta. Observer design for 2-D discrete systems with delays. *International Conference on Control and Automation*, Budapest, Hungary, 2005.

[14] E.I. Jury, P. Bauer. On the Stability of Two-Dimensional Continuous Systems. *IEEE Transactions on Circuits and Systems*, 35 (12), 1988.

[15] T. Kaczorek. General response formulae for two-dimensional linear systems with variable coefficients. *IEEE Transactions on Automatic Control*, 31 (3), 278-280, 1986.

[16] S. Knorn. *A two-dimensional systems stability analysis of vehicle platoons*. Ph.D. dissertation, National University of Ireland, Maynooth, 2013.

[17] J.E. Kurek. Stability of nonlinear time-varying digital 2-D Fornasini-Marchesini system. *Multidimensional Systems and Signal Processing*, 25, 235-244, 2014.

[18] J.E. Kurek. Observability and reconstructability of the 2-D Fornasini-Marchesini model. *IEEE Transactions on Circuits and Systems*, 35 (8), 1011-1014, 1988.

[19] F.L. Lewis. A review of 2-D implicit systems. *Automatica*, 28 (2), 345-354, 1992.

[20] Y. Luo, Z. Wang, J. Hu, G. Wei. H_∞ gain-scheduling control for 2-D stochastic nonlinear systems: The F-M case. *Chinese Control Conference*, Chengdu, China, 2016.

[21] W.S. Lu. Some new results on stability and stability robustness of Fornasini-Marchesini stat-space 2-D digital filters. *IEEE Creative Technology Transfer-A Global Affair*, Miami, FL, USA, 1994.

[22] H. Luo, W.S. Lu, A. Antoniou. New algorithms for the derivation of the transfer-function matrices of 2-D state-space discrete systems. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 44 (2), 112-119, 1997.

[23] B.G. Mertzios, P.N. Paraskevopoulos. Transfer function matrix of 2-D systems. *IEEE Transactions on Automatic. Control*, 26(3), 722-724. 1981.

[24] L. Ntogramatzidis, M. Cantoni. LQ optimal control for 2-D Roesser models of finite extent. *Systems and Control Letters*, 58 (7), 482-490, 2009.

[25] D.H. Owe, E. Rogers. Boundary Conditions and the Stability of a Class of 2-D Continuous-discrete Linear Systems. *American Control Conference*, Portland, OR, USA, 2005.

[26] R.J. Patton, P.M. Frank, R.N. Clark, *Issues of fault diagnosis for dynamical systems*, Springer, 2000.

[27] J. Ragot, A. Aïtouche, F. Kratz, D. Maquin. Detection and location of gross errors in instruments using parity space technique. *International Journal of Mineral Processing*, Elsevier, 1991, 31 (3-4), 281-299, 1991.

[28] R. Roesser. Discrete state-space model for linear image processing. *IEEE Transactions on Automatic Control*, 20 (1), 1-10, 1975.

[29] Y. Wang, H. Zhang, S. Wei, D. Zhou, B. Huang. Control Performance Assessment for ILC-Controlled Batch Processes in a 2-D System Framework. *IEEE Transactions on Systems, Man and Cybernetics: Systems*, 99, 2017.

[30] R. Whalley, Z. Zeng. Algorithms for two-dimensional transfer function computation. *Applied Mathematical Modelling*, 16 (12), 659-665, 1992.

[31] M. Yamada, L. Xu, O. Saito. 2-D Model-Following Servo System. *Multidimensional Systems and Signal Processing*, 10, 71-91, 1999.

[32] E. Yaz. New approach to observer design for two-dimensional systems. *International Journal of Control*, 45 (1), 355-358, 1987.