State estimation and fault detection for 2-D discrete-time systems
Dalil Ichalal, Benoît Marx, José Ragot

To cite this version:
Dalil Ichalal, Benoît Marx, José Ragot. State estimation and fault detection for 2-D discrete-time systems. 26th Mediterranean Conference on Control and Automation, MED’18, Jun 2018, Zadar, Croatia. 10.1109/MED.2018.8442908 . hal-01777593

HAL Id: hal-01777593
https://hal.science/hal-01777593
Submitted on 25 Nov 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Abstract—This paper presents an observer based sensor fault detection 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 bank 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], $H_\infty$ 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]. Among 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:

$$
\begin{align*}
\begin{cases}
x(i+1, j+1) = A_1 x(i+1, j) + A_2 x(i, j+1) + B u(i, j) \\
y(i, j) = C x(i, j)
\end{cases}
\end{align*}
$$

(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:

$$
W(z_1, z_2) = C(I - A_1 z_1 - A_2 z_2)^{-1} B z_1 z_2
$$

(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.

![State calculus scheme](image)

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:

$$
\begin{align*}
\begin{cases}
A^{(i,j)} = A_1 A^{(i-1,j)} + A_2 A^{(i,j-1)} \\
A^{(0,0)} = I \\
A^{(i,j)} = 0 & \text{if } i < 0 \text{ or } & j < 0
\end{cases}
\end{align*}
$$

(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 $j \geq k$. 

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

2Benoît 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 
\((C,A_1r + A_2s)\) is reconstructible for all \(r,s \in C\).

### 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:

\[
x(i,j) = \sum_{k=1}^{j} A^{(j-k,i-1)} A_2 x(0,k) + \sum_{k=1}^{i} A^{(j-1,i-k)} A_1 x(k,0) + \sum_{n=0}^{j-1} \sum_{i=0}^{j} A^{(j-n-1,i-m-1)} Bu(m,n)
\]

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

\[
y(i,j) = \sum_{k=1}^{j} CA^{(j-k,i-1)} A_2 x(0,k) + \sum_{k=1}^{i} CA^{(j-1,i-k)} A_1 x(k,0) + \sum_{n=0}^{j-1} \sum_{i=0}^{j} CA^{(j-n-1,i-m-1)} Bu(m,n)
\]

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.

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

\[
x(1,1) = A_1 x(1,0) + A_2 x(0,1) + Bu(0,0)
\]

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:

\[
x(i+1,j) = A_1 x(i+1,j-1) + A_2 x(i,j) + Bu(i,j-1)
\]

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

\[
x(i+1,j) = A_1 \left( \sum_{n=0}^{j-2} \sum_{i=0}^{j} A^{(j-n-2,i-m)} Bu(m,n) + \sum_{k=1}^{j} A^{(j-1-k,i)} A_2 x(0,k) + \sum_{k=1}^{j} A^{(j-2,i+1-k)} A_1 x(k,0) \right) + Bu(i,j-1) + A_2 \left( \sum_{n=0}^{j-1} \sum_{i=0}^{j} A^{(j-n-1,i-m-1)} Bu(m,n) + \sum_{k=1}^{j} A^{(j-1-k,i-1)} A_2 x(0,k) + \sum_{k=1}^{j} A^{(j-1,i-k)} A_1 x(k,0) \right)
\]

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

\[
p_1 = A_1 \left( \sum_{k=1}^{j} A^{(j-1-k,i)} A_2 x(0,k) \right) + A_2 \left( \sum_{k=1}^{j} A^{(j-k,i-1)} A_2 x(0,k) \right) = \sum_{k=0}^{j} (A_1 A^{(j-1-k,i)} + A_2 A^{(j-k,i-1)}) A_2 x(0,k)
\]

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

\[
p_2 = A_1 \sum_{k=1}^{j} A^{(j-2,i+1-k)} A_1 x(k,0) + A_2 \sum_{k=1}^{j} A^{(j-1,i-k)} A_1 x(k,0) = \sum_{k=1}^{j} A^{(j-2,i+1-k)} A_1 x(k,0)
\]

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

\[
p_3 = A_1 \sum_{n=0}^{j} \sum_{i=0}^{j} A^{(j-n-2,i-m)} Bu(m,n) + A_2 \sum_{n=0}^{j} \sum_{i=0}^{j} A^{(j-n-1,i-m-1)} Bu(m,n) + Bu(i,j-1) = \sum_{m=0}^{j} \sum_{i=0}^{j} A^{(j-n-1,i-m)} Bu(m,n)
\]

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

\[
x(i+1,j) = \sum_{k=0}^{j} A^{(j-k,i)} A_2 x(0,k) + \sum_{k=1}^{j+1} A^{(j-2,i+1-k)} A_1 x(k,0) + \sum_{m=0}^{j-1} \sum_{i=0}^{j} A^{(j-n-1,i-m)} Bu(m,n)
\]
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 \( x(0, j), j = 1, \ldots, J \) and \( x(i, 0), i = 1, \ldots, I \).

By stating the equations in the interval \((i, j) \in [1: J] \times [1: J]\), 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:

\[
\begin{align*}
  x &= F_x x_0 + F_u u \\
  y &= G_x x_0 + G_u u
\end{align*}
\]  

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

\[
\begin{align*}
  x &= \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_I \end{bmatrix}, \\
  x_0 &= \begin{bmatrix} x(i, 1) \\ x(i, 2) \\ \vdots \\ x(i, J) \end{bmatrix}, \\
  u &= \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{J-1} \end{bmatrix}, \\
  u_i &= \begin{bmatrix} u(i, 0) \\ u(i, 1) \\ \vdots \\ u(i, J-1) \end{bmatrix}
\end{align*}
\]

where \( x \in \mathcal{R}^{J \times n_x}, u_i \in \mathcal{R}^{I \times n_u}, F_x, F_u, G_x \) 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:

\[
\Phi = \| y - G_x x_0 - G_u u \|^2_W
\]  

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:

\[
\hat{x}_0 = (G^T_x W G_x)^{-1} G^T_x W (y - G_u u)
\]  

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

\[
\begin{align*}
  \dot{x} &= F_x \hat{x}_0 + F_u u \\
  \hat{y} &= G_x \hat{x}_0 + G_u u
\end{align*}
\]

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

\[
\begin{align*}
  x(i + 1, j + 1) &= A_1 x(i + 1, j) + A_2 x(i, j + 1) + B u(i, j) \\
  y(i, j) &= C x(i, j) + f
\end{align*}
\]

where \( f = [f_1 \ldots f_n]^T \) is the sensor fault. The output estimation of the FMO (17) allows to generate the output reconstruction error: \( \hat{y}(i, j) = y(i, j) - \dot{y}(i, j) \) that is a sensor fault indicator or residual. Its analysis allows to perform 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 \([0 : 2] \times [0 : 2]\),

\[
\begin{bmatrix} y(1, 0) \\ y(2, 0) \\ y(0, 1) \\ y(0, 2) \\ y(1, 1) \\ y(1, 2) \\ y(2, 1) \\ y(2, 2) \end{bmatrix} = C \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ C \\ 0 \\ 0 \\ 0 \end{bmatrix}
\]

\[
A_1 \begin{bmatrix} 0 \\ C \end{bmatrix} + A_2 \begin{bmatrix} 0 \\ 0 \end{bmatrix}
\]

\[
CA_1 \begin{bmatrix} 0 \\ 0 \end{bmatrix} + CA_2 \begin{bmatrix} C \\ 0 \end{bmatrix}
\]

\[
CA_{(1,1)} A_1 \begin{bmatrix} C \\ 0 \end{bmatrix} + CA_{(1,2)} A_2 \begin{bmatrix} 0 \\ C \end{bmatrix}
\]  

\[
\begin{bmatrix} x(1, 0) \\ x(2, 0) \\ x(0, 1) \\ x(0, 2) \end{bmatrix}
\]

\[
\begin{bmatrix} u(0, 0) \\ u(0, 1) \\ u(1, 0) \\ u(1, 1) \end{bmatrix}
\]  

\[
\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}
\]  

\[
\begin{bmatrix} C \times (1, 0) \\ C \times (0, 0) \end{bmatrix}
\]

\[
\begin{bmatrix} C \times (1, 1) \\ C \times (1, 2) \end{bmatrix}
\]
In (19) structure of (13) is preserved with $G_x$ and $G_u$ defined by:

$$G_x = \begin{bmatrix} C & 0 & 0 & 0 \\ 0 & C & 0 & 0 \\ 0 & 0 & C & 0 \end{bmatrix}$$

and

$$G_u = \begin{bmatrix} CA_1 & 0 & CA_2 & 0 \\ 0 & CA_1 & 0 & CA_2 \\ CA_2 A_1 & CA_1 & CA_2 & 0 \\ CA^{(1,1)} A_1 & CA_2 & CA^{(1,1)} A_2 & CA_2 \end{bmatrix}$$

The particular structure of $G_x$ can be used to study its rank, that is useful to analyse the rank of $G_x^T W G_x$ 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:

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

The input control is given by: $u = \sin(t/15 \cos(t/17))/2 + 0.1$. The states $x(i, j) \in \mathbb{R}^3$ 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 outputs $\{y_1, y_2, y_3\}$ 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 $\tilde{y}(i, j) = y(i, j) - C \hat{x}(i, j)$.

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

$$f_1 \quad f_2 \quad f_3$$

<table>
<thead>
<tr>
<th>$\tilde{y}_{(y_1, y_2, y_3)}$</th>
<th>$\times$</th>
<th>$\times$</th>
<th>$\times$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tilde{y}_{(y_1, y_2)}$</td>
<td>$\times$</td>
<td>$\times$</td>
<td>$\times$</td>
</tr>
<tr>
<td>$\tilde{y}_{(y_1, y_3)}$</td>
<td>$\times$</td>
<td>$\times$</td>
<td>$\times$</td>
</tr>
<tr>
<td>$\tilde{y}_{(y_2, y_3)}$</td>
<td>$\times$</td>
<td>$\times$</td>
<td>$\times$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TABLE I</th>
</tr>
</thead>
<tbody>
<tr>
<td>THEORETICAL FAULT SIGNATURE</td>
</tr>
</tbody>
</table>
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.

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


