Parity space-based fault detection for Markovian jump systems*

Maiying Zhong[†], Qiang Ding[‡], Peng Shi[§]

Abstract. This paper deals with problems of parity space-based fault detection for a class of discrete-time linear Markovian jump systems. A new algorithm is firstly introduced to reduce the computation of mode-dependent redundancy relation parameter matrices. Different from the case of linear time invariant systems, the parity space-based residual generator for a Markovian jump system cannot be designed off-line because it depends on the history of system modes in the last finite steps. In order to overcome this difficulty, a finite set of parity matrices is pre-designed applying a unified approach to linear time invariant systems. Then the on-line residual generation can be easily implemented. Moreover, the problem of residual evaluation is also considered which includes the determination of a residual evaluation function and a threshold. Finally, a numerical example is given to illustrate the effectiveness of the proposed method.

Keywords. Fault detection, Markovian jump system, residual, parity space, threshold.

1 Introduction

Due to the increasing demand for safety and reliability of technique process, research on model-based fault detection and isolation (FDI) has received considerable attention over the past 30 years, see for example, [1, 5, 6, 7, 9, 11, 12] and references therein. A model-based FDI system is a dynamic system which takes the input and measurement output of the process as input signals and delivers alarm information about fault when plant, actuator or sensor faults occur. Parity space-based fault detection is a time domain approach which constructs a residual generator by collecting a batch of control input and measurement output data within a window of certain length and, based on this, to formulate the design of residual generator as an optimal selection of parity vector (or matrix). As is well known, the realization of parity space-based approach

*The work was partially supported by the NSF of China (60774071,60736025), the PhD Program Foundation of Education Ministry of China (20050422036), and the Grant of Shandong Province (2005BS01007).

[†]All correspondence should be addressed to Maiying Zhong, myzhong@sdu.edu.cn. School of Instrument Science and Opto-Electronics Engineering, Beihang University, 37 Xueyuan Road, 100083 Beijing, China.

[‡]School of Control Science and Engineering, Shandong University, 250061 Jinan, China.

[§]Faculty of Advanced Technology, University of Glamorgan, UK; School of Computer Science and Mathematics, Victoria University, Australia; and School of Mathematics and Statistics, University of South Australia.

involves only solutions of algebraic equations, and system state is completely decoupled from residual. So, it is a popular approach for general linear time invariant systems and has been studied extensively by [2, 3], and [4]. For time-varying systems, however, it is not an easy task to realize parity space-based FDI due to the on-line numerical computation of parity space equation coefficient matrices.

On the other hand, Markovian jump system is one of the time-varying systems of which one state takes values discretely in a finite set. This kind of system can be used to represent many important physical systems subject to random failures and structure changes, such as electric power systems, control systems of a solar thermal central receiver, communication systems, aircraft flight control, control of nuclear power plants, manufacturing systems, and networked control systems. In the past decades, the control and filtering problems have been extensively studied for Markovian jump systems, see [8, 15, 13] and references therein.

Associated with the increasing demands for system safety and reliability, FDI for Markovian jump system has been one of the most critical issues surrounding system analysis and design. More recently, a few achievements have been published in the literature [10, 14, 17, 18]. In [17], an observer-based residual generator has been designed by solving a two-objective optimization problem; in [14] and [18], the problems of observer-based fault detection are considered in the framework of $\mathcal{H}_\infty$-filtering formulation; in [10], a networked control system was modelled by a discrete-time Markovian jump system and the problem of $\mathcal{H}_\infty$ fault detection filter was designed. To the best of authors' knowledge, however, the problem of parity space-based fault detection for Markovian jump systems has not been fully investigated yet, which motivates us for the present study.

In this paper, we will study the problem of parity space-based fault detection for a class of discrete-time linear Markovian jump systems. Our attention is focused on the development of a new algorithm to reduce the numerical computation of redundancy relation parameter matrices. Then, a finite set of parity matrices will be designed by extending a unified parity space-based approach to LTI systems. Moreover, the on-line residual generation for Markovian jump systems will be implemented. Finally, the problem of residual evaluation will be considered to determine a residual evaluation function and a threshold. To demonstrate the feasibility and effectiveness of the obtained results, a numerical example is also included.

2 Problem formulation

Consider the following class of Markovian jump systems

$$\left\{ \begin{array}{r} x(k+1) = A(\theta_k)x(k) + B(\theta_k)u(k) + B_d(\theta_k)d(k) + B_f(\theta_k)f(k) \\ y(k) = C(\theta_k)x(k) + D_d(\theta_k)d(k) + D_f(\theta_k)f(k) \end{array} \right. \tag{1}$$

for $k = 0, 1, 2, \ldots$, where $x(k) \in \mathbb{R}^n$ is the state, $y(k) \in \mathbb{R}^q$ the output, $u(k) \in \mathbb{R}^m$ the known input, $d(k) \in \mathbb{R}^p$ the unknown input, $f(k) \in \mathbb{R}^l$ the fault to be detected; $\{\theta_k\}$ is a discrete homogeneous Markov chain taking values in a finite state space $\Omega = \{1, 2, \ldots, N\}$ with transition probability matrix $\Lambda = [\lambda_{ij}]_{i,j\in\Omega}$, and λ_{ij} is defined as

$$\lambda_{ij} = \Pr\{\theta_{k+1} = j \mid \theta_k = i\}$$

and $\sum_{j=1}^{N} \lambda_{ij} = 1$; $A(\theta_k)$, $B(\theta_k)$, $B_d(\theta_k)$, $B_f(\theta_k)$, $C(\theta_k)$, $D_d(\theta_k)$ and $D_f(\theta_k)$ are known real constant matrices for all $\theta_k = i \in \Omega$. Denote the matrices associated with $\theta_k = i \in \Omega$ by

$$\begin{aligned} A(\theta_k) &= A_i,\ B(\theta_k) = B_i,\ B_d(\theta_k) = B_{di},\ B_f(\theta_k) = B_{fi} \\ C(\theta_k) &= C_i,\ D_d(\theta_k) = D_{di},\ D_f(\theta_k) = D_{fi} \end{aligned}$$

We begin with the parity space-based residual generator for system (1). For a given integer $s > 0$, combining together (1) from time instant $k-s$ to k yields the following redundancy equation

$$y_s(k) - H_{us}(\theta_k)u_s(k) = H_{os}(\theta_k)x(k-s) + H_{ds}(\theta_k)d_s(k) + H_{fs}(\theta_k)f_s(k)$$

where

$$\begin{aligned} y_s(k) &= \begin{bmatrix} y(k-s) \\ y(k-s+1) \\ \vdots \\ y(k) \end{bmatrix},\ u_s(k) = \begin{bmatrix} u(k-s) \\ u(k-s+1) \\ \vdots \\ u(k) \end{bmatrix} \\ d_s(k) &= \begin{bmatrix} d(k-s) \\ d(k-s+1) \\ \vdots \\ d(k) \end{bmatrix},\ f_s(k) = \begin{bmatrix} f(k-s) \\ f(k-s+1) \\ \vdots \\ f(k) \end{bmatrix} \end{aligned}$$

$$H_{os}(\theta_k) = \begin{bmatrix} C(\theta_{k-s}) \\ C(\theta_{k-s+1})A(\theta_{k-s}) \\ \vdots \\ C(\theta_{k-1})A(\theta_{k-2})\cdots A(\theta_{k-s+1})A(\theta_{k-s}) \\ C(\theta_k)A(\theta_{k-1})\cdots A(\theta_{k-s+1})A(\theta_{k-s}) \end{bmatrix} \tag{2}$$

$$H_{ds}(\theta_k) = \begin{pmatrix} D_d(\theta_{k-s}) & 0 & \cdots & \cdots & 0 \\ (2,1) & D_d(\theta_{k-s+1}) & \ddots & & \vdots \\ (3,1) & (3,2) & D_d(\theta_{k-s+2}) & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ (s+1,1) & \cdots & (s+1,s-1) & (s+1,s) & D_d(\theta_k) \end{pmatrix} \tag{3}$$

and

$$
\begin{aligned}
(2,1) &= C(\theta_{k-s+1})B_d(\theta_{k-s}) \\
(3,1) &= C(\theta_{k-s+2})A(\theta_{k-s+1})B_d(\theta_{k-s}) \\
(3,2) &= C(\theta_{k-s+2})B_d(\theta_{k-s+1}) \\
(s+1,1) &= C(\theta_k)A(\theta_{k-1})\cdots A(\theta_{k-s+1})B_d(\theta_{k-s}) \\
(s+1,s-1) &= C(\theta_k)A(\theta_{k-1})B_d(\theta_{k-2}) \\
(s+1,s) &= C(\theta_k)B_d(\theta_{k-1})
\end{aligned}
$$

The matrix $H_{us}(\theta_k)$ is constructed by replacing B_d and D_d with B and 0 in (3), respectively; similarly, the matrix $H_{fs}(\theta_k)$ is constructed by replacing B_d and D_d with B_f and D_f, respectively.

For the purpose of fault detection, the following parity space-based residual generator is defined as

$$r(\theta_k) = W_s(\theta_k)(y_s(k) - H_{us}(\theta_k)u_s(k)) \tag{4}$$

where $r(\theta_k)$ is the residual vector, $W_s(\theta_k)$ is a mode-dependent matrix satisfying

$$W_s(\theta_k)H_{os}(\theta_k) = 0, \forall\theta_k = i \in \Omega$$

which is refereed to parity matrix. Denoting the parity space of order s by

$$\mathcal{P}_s(\theta_k) = \{W_s(\theta_k)\,|W_s(\theta_k)H_{os}(\theta_k) = 0\}$$

we then have

$$r(\theta_k) = W_s(\theta_k)(H_{ds}(\theta_k)d_s(k) + H_{fs}(\theta_k)f_s(k))$$

It is usually not an easy task to update $H_{os}(\theta_k)$, $H_{us}(\theta_k)$, $H_{ds}(\theta_k)$, and $H_{fs}(\theta_k)$ at every time instant k due to the numerical matrix operations. Notice that, however, there are lots of repeated operations in $H_{\varsigma s}(\theta_{k-1})$ and $H_{\varsigma s}(\theta_k)$, where ς stands for o, u, d, or f in sequence. Therefore, we first need to develop a recursive algorithm for reducing the on-line calculation of $H_{\varsigma s}(\theta_k)$, and design residual generator next. In this paper, we propose to pre-design the finite points of parity matrices $W_s(\theta_k)$, $\forall\theta_k = i \in \Omega$, applying the unified approach in [4]. Without loss of generality, it is assumed that the calculation of $H_{os}(\theta_k)$, $H_{ds}(\theta_k)$, $H_{fs}(\theta_k)$, $N_b(\theta_k)$, and $P_s(k)$ can be completed within time step $[k-1, k]$.

Denote the dimension and the basis matrix of parity space $\mathcal{P}_s(\theta_k)$ by γ_{ki} and $N_{bi}(k)$, respectively, and let

$$H_{ds,i}(k) = H_{ds}(\theta_k),\ H_{fs,i}(k) = H_{fs}(\theta_k),\ P_{si}(k)N_{bi}(k) = W_{si}(k) = W_s(\theta_k)$$

Then the residual generator corresponding to $\theta_k = i \in \Omega$ is governed by

$$r_i(k) = P_{si}(k)N_{bi}(k)(H_{ds,i}(k)d_s(k) + H_{fs,i}(k)f_s(k))$$

We now formulate the problem of residual generation as to find matrix $P_s(\theta_k)$ such that, $\forall\theta_k = i \in \Omega$, the matrix $P_{si}(k) = P_s(\theta_k)$ solves the following minimization problem

$$\min_{P_{si}(k)} \frac{\|P_{si}(k)N_{bi}(k)H_{ds,i}(k)\|}{\|P_{si}(k)N_{bi}(k)H_{fs,i}(k)\|},\ k = k_0, k_0+1, \cdots \tag{5}$$

where $k_0 \geq s$ is the initial time step.

After designing of residual generator, the remaining important task of fault detection is the residual evaluation which concerns with the selection of residual evaluation function $J_r(k)$ and the determination of a corresponding threshold $J_{th}(k)$. Thus the following logic unit can be used to detect the occurrence of a fault

$$\begin{cases} J_r(k) > J_{th}(k) & \Rightarrow \text{fault alarm} \\ J_r(k) \leq J_{th}(k) & \Rightarrow \text{no fault} \end{cases} \tag{6}$$

We now restate our objectives in this paper: (i) develop a recursive algorithm for reducing the on-line computation of $H_{os}(\theta_k),\ H_{us}(\theta_k),\ H_{ds}(\theta_k)$, and $H_{fs}(\theta_k)$; (ii) solve minimization problem (5); and (iii) determine $J_r(k)$ and $J_{th}(k)$.

3 Main results

In this section, we will first develop a new algorithm for reducing the computation of $H_{os}(\theta_k),\ H_{us}(\theta_k)$, $H_{ds}(\theta_k)$ and $H_{fs}(\theta_k)$.

Re-express $H_{ds}(\theta_k)$ as

$$H_{ds}(\theta_k) = \begin{bmatrix} D_d(\theta_{k-s}) & 0 \\ * & H_{ds,2}(\theta_k) \end{bmatrix} \tag{7}$$

where

$$H_{ds,2}(\theta_k) = \begin{bmatrix} D_d(\theta_{k-s+1}) & 0 & \cdots & \cdots & 0 \\ (2,1) & D_d(\theta_{k-s+2}) & \ddots & & \vdots \\ (3,1) & (3,2) & D_d(\theta_{k-s+3}) & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ (s,1) & \cdots & (s,s-2) & (s,s-1) & D_d(\theta_k) \end{bmatrix} \tag{8}$$

with

$$\begin{aligned} (2,1) &= C(\theta_{k-s+2})B_d(\theta_{k-s+1}) \\ (3,1) &= C(\theta_{k-s+3})A(\theta_{k-s+2})B_d(\theta_{k-s+1}) \\ (3,2) &= C(\theta_{k-s+3})B_d(\theta_{k-s+2}) \\ (s,1) &= C(\theta_k)A(\theta_{k-1})\cdots A(\theta_{k-s+2})B_d(\theta_{k-s+1}) \\ (s,s-2) &= C(\theta_k)A(\theta_{k-1})B_d(\theta_{k-2}) \\ (s,s-1) &= C(\theta_k)B_d(\theta_{k-1}) \end{aligned}$$

and “$*$” denote terms of no interest. For given $H_{ds}(\theta_{k-1})$, we then update $H_{ds}(\theta_k)$ using

$$H_{ds}(\theta_k) = \begin{bmatrix} H_{ds,2}(\theta_{k-1}) & 0 \\ \bar{H}_{ds,2}(\theta_k) & D_d(\theta_k) \end{bmatrix} \tag{9}$$

where

$$\bar{H}_{ds,2}(\theta_k) \;=\; C(\theta_k)\begin{bmatrix} A(\theta_{k-1})\cdots A(\theta_{k-s+1})B_d(\theta_{k-s}) & \cdots & A(\theta_{k-1})B_d(\theta_{k-2}) & B_d(\theta_{k-1}) \end{bmatrix}$$

Defining

$$\begin{aligned}
B_{ds}(\theta_k) &= \operatorname{diag}\left(B_d(\theta_{k-s+1}), \ \cdots, \ B_d(\theta_{k-1}), \ B_d(\theta_k) \right) \\
\Gamma(\theta_k, 1) &= A(\theta_k)A(\theta_k - 1)\cdots A(\theta_{k-s+2}) \\
\Gamma(\theta_k, 2) &= A(\theta_k)A(\theta_k - 1)\cdots A(\theta_{k-s+3}) \\
&\ \vdots \\
\Gamma(\theta_k, s-1) &= A(\theta_k) \\
\Gamma(\theta_k, s) &= I \\
\Gamma_1(\theta_k) &= \begin{bmatrix} \Gamma(\theta_k, 1) & \Gamma(\theta_k, 2) & \cdots & \Gamma(\theta_k, s-1) \end{bmatrix} \\
\Gamma(\theta_k) &= \begin{bmatrix} \Gamma(\theta_k, 1) & \Gamma(\theta_k, 2) & \cdots & \Gamma(\theta_k, s) \end{bmatrix}
\end{aligned}$$

then $\bar{H}_{ds,2}(\theta_k)$ and $\Gamma(\theta_k)$ can be rewritten as

$$\bar{H}_{ds,2}(\theta_k) = C(\theta_k)\Gamma(\theta_{k-1})B_{ds}(\theta_{k-1}) \tag{10}$$

$$\Gamma(\theta_k) = \begin{bmatrix} A(\theta_k)\Gamma_1(\theta_{k-1}) & I \end{bmatrix} \tag{11}$$

Therefore, for given $H_{ds}(\theta_{k-1})$ and $\Gamma(\theta_k)$, one can obtain $H_{ds}(\theta_k)$ by (9), (10), and (11), which is summarized into the following Algorithm 1.

Algorithm 1:

Step 1. Get $H_{ds,2}(\theta_{k-1})$ from $H_{ds}(\theta_{k-1})$ in (7), i.e.

$$H_{ds,2}(\theta_{k-1}) = \begin{bmatrix} 0 & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} D_d(\theta_{k-s-1}) & 0 \\ * & H_{ds}(\theta_{k-1}) \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & I \end{bmatrix}$$

Step 2. Get $\Gamma_1(\theta_{k-1})$ from $\Gamma(\theta_{k-1})$ according to $\Gamma(\theta_{k-1}) = \begin{bmatrix} \Gamma_1(\theta_{k-1}) & I \end{bmatrix}$;

Step 3. Compute $\bar{H}_{ds,2}(\theta_k)$ by (10)–(11);

Step 4. Obtain $H_{ds}(\theta_k)$ from (9).

Remark 1 *It is noted from Algorithm 1 that, for given $\Gamma(\theta_{k-1})$ and $H_{ds}(\theta_{k-1})$, $\bar{H}_{ds,2}(\theta_k)$ takes the main computation cost of $H_{ds}(\theta_k)$, which needs the s times $\mathbb{R}^{n\times n} \times R^{n\times p}$, one time $\mathbb{R}^{q\times n} \times R^{n\times n}$, and s times $\mathbb{R}^{q\times n} \times R^{n\times p}$ matrix operations. If one calculates $H_{ds}(\theta_k)$ by (3) directly, then the $\frac{1}{2}(s+1)(s+2)$ nonzero terms of $H_{ds}(\theta_k)$ should be updated at every time step k. In* *order to update terms* $(2,1),\ (3,2), \cdots, (s+1,s)$, s *times* $\mathbb{R}^{q\times n} \times R^{n\times p}$ *matrix operations are necessary; terms* $(3,1),\ (4,2),\ \cdots, (s+1,s-1)$ *need* $s-1$ *times* $\mathbb{R}^{q\times n} \times R^{n\times n}$ *and* $s-1$ *times* $\mathbb{R}^{q\times n} \times R^{n\times p}$ *matrix operations; term* $(s+1,1)$ *needs* $s-1$ *times* $\mathbb{R}^{q\times n} \times R^{n\times n}$ *and one time* $\mathbb{R}^{q\times n} \times R^{n\times p}$ *matrix operations. From the analysis we can see that the computation cost of* $H_{ds}(\theta_k)$ *using (9), (10) and (11) is less than that of using (3).*

Similarly, $H_{us}(\theta_k)$ and $H_{fs}(\theta_k)$ can be updated using

$$H_{us}(\theta_k) = \begin{bmatrix} H_{us,2}(\theta_{k-1}) & 0 \\ \bar{H}_{us,2}(\theta_k) & 0 \end{bmatrix} \text{ and } H_{fs}(\theta_k) = \begin{bmatrix} H_{fs,2}(\theta_{k-1}) & 0 \\ \bar{H}_{fs,2}(\theta_k) & D_f(\theta_k) \end{bmatrix} \tag{12}$$

respectively, where

$$\bar{H}_{us,2}(\theta_k) = C(\theta_k)\Gamma(\theta_{k-1})B_s(\theta_{k-1}),\ \bar{H}_{fs,2}(\theta_k) = C(\theta_k)\Gamma(\theta_{k-1})B_{fs}(\theta_{k-1}) \tag{13}$$

$$B_s(\theta_k) = \operatorname{diag}\left(\begin{array}{cccc} B(\theta_{k-s+1}), & \cdots, & B(\theta_{k-1}), & B(\theta_k) \end{array}\right) \tag{14}$$

$$B_{fs}(\theta_k) = \operatorname{diag}\left(\begin{array}{cccc} B_f(\theta_{k-s+1}), & \cdots, & B_f(\theta_{k-1}), & B_f(\theta_k) \end{array}\right) \tag{15}$$

$H_{uk,2}(\theta_{k-1})$ and $H_{fk,2}(\theta_{k-1})$ are entries of

$$H_{us}(\theta_{k-1}) = \begin{bmatrix} 0 & 0 \\ * & H_{us,2}(\theta_{k-1}) \end{bmatrix} \text{ and } H_{fs}(\theta_{k-1}) = \begin{bmatrix} D_f(\theta_{k-s-1}) & 0 \\ * & H_{fs,2}(\theta_{k-1}) \end{bmatrix}$$

respectively.

Defining

$$C_s(\theta_k) = \operatorname{diag}\left(\begin{array}{cccc} C(\theta_{k-s}), & \cdots, & C(\theta_{k-1}), & C(\theta_k) \end{array}\right)$$

$$\Upsilon(\theta_k) = \begin{bmatrix} I \\ A(\theta_{k-s+1}) \\ \vdots \\ A(\theta_{k-1})\cdots A(\theta_{k-s+2})A(\theta_{k-s+1}) \\ A(\theta_k)\cdots A(\theta_{k-s+2})A(\theta_{k-s+1}) \end{bmatrix}$$

it follows from (2) that

$$H_{os}(\theta_k) = C_s(\theta_k)\Upsilon(\theta_{k-1}) \tag{16}$$

$$\Upsilon(\theta_k) = \begin{bmatrix} I \\ \Gamma(\theta_{k-s+1}, s-1) \\ \Gamma(\theta_{k-s+2}, s-2) \\ \vdots \\ \Gamma(\theta_{k-1}, 1) \\ A(\theta_k)\Gamma(\theta_{k-1}, 1) \end{bmatrix} \tag{17}$$

Observing that $\Gamma(\theta_{k-i})$ $(i = 1, 2, \cdots, s-1)$ are known at time step k and $\Gamma(\theta_{k-i}, i)$ is an entry of $\Gamma(\theta_{k-i})$, we can get $H_{os}(\theta_k)$ by computing $A(\theta_k)\Gamma(\theta_{k-1}, 1)$ and $C_s(\theta_k)\Upsilon(\theta_{k-1})$ according to (16)–(17), which need one time $\mathbb{R}^{n\times n} \times R^{n\times n}$ and s times $\mathbb{R}^{q\times n} \times R^{n\times n}$ matrix operations. If $H_{os}(\theta_k)$ is calculated by (2) directly, then one needs $\frac{1}{2}s(s+1)$ times $\mathbb{R}^{q\times n} \times R^{n\times n}$ matrix operations. Therefore, the computation cost of $H_{os}(\theta_k)$ using (16)–(17) is less than that of using (2).

From the above analysis, we have the following Proposition 1.

Proposition 1 *The computation of $H_{us}(\theta_k)$, $H_{ds}(\theta_k)$, $H_{fs}(\theta_k)$ and $H_{os}(\theta_k)$ can be reduced by using recursive formulas (9)–(11), (12)–(15) and (16)–(17).*

Next, we consider to solve the minimization problem (5) for $\theta_k = i \in \Omega$. If $N_{bi}(k)H_{ds,i}(k)$ is not full-row rank, i.e.

$$\beta_{ki} = \mathrm{rank}(N_{bi}(k)H_{ds,i}(k)) < \gamma_{ki}$$

then there exists $P_{si}(k) \in \mathbb{R}^{(\gamma_{ki}-\beta_{ki})\times\gamma_{ki}}$ such that $P_{si}(k)N_{bi}(k)H_{ds,i}(k) = 0$. Thus, a solution to the minimization problem (5) can be obtained by solving $P_{si}(k)N_{bi}(k)H_{ds,i}(k) = 0$ and, in this case, we have

$$\min_{P_{si}(k)} \frac{\|P_{si}(k)N_{bi}(k)H_{ds,i}(k)\|}{\|P_{si}(k)N_{bi}(k)H_{fs,i}(k)\|} = 0$$

which means that unknown input $d_s(k)$ is completely decoupled from residual $r_i(k)$ at k.

If $N_{bi}(k)H_{ds,i}(k)$ being full-row rank for $\theta_k = i$, i.e. $\mathrm{rank}(N_{bi}(k)H_{ds,i}(k)) = \gamma_{ki}$, one can solve the minimization problem (5) by applying the unified approach in [4], which is summarized in the following Proposition 2.

Proposition 2 *Suppose that* $\mathrm{rank}(N_{bi}(k)H_{ds,i}(k)) = \gamma_{ki}$*,* $N_{bi}(k)H_{ds,i}(k)H_{ds,i}^T(k)N_{bi}^T(k)$ *has a singular value decomposition (SVD) given by*

$$N_{bi}(k)H_{ds,i}(k)H_{ds,i}^T(k)N_{bi}^T(k) = U_{ki}\Sigma_{ki}U_{ki}^T,\ U_{ki}^TU_{ki} = I$$
$$\Sigma_{ki} = \mathrm{diag}(\sigma_{ki,1}, \sigma_{ki,2}, \cdots, \sigma_{ki,\gamma_{ki}}),\ \sigma_{ki,1} \geq \sigma_{ki,2} \geq \cdots \geq \sigma_{ki,\gamma_{ki}} > 0$$

then $P_{si}^*(k) = \Sigma_{ki}^{-\frac{1}{2}}U_{ki}^T$ solves the minimization problem (5) and

$$\frac{\|P_{si}^*(k)N_{bi}(k)H_{ds,i}(k)\|}{\|P_{si}^*(k)N_{bi}(k)H_{fs,i}(k)\|} = \frac{1}{\delta_{ki,1}}$$

where $\delta_{ki,1}$ is the largest singular value of $\Sigma_{ki}^{-\frac{1}{2}}U_{ki}^TN_{bi}(k)H_{fs,i}(k)$.

Remark 2 *In [19], Section 2.6 provided with us a review to singular value decomposition. Using the introduced SVD method one can get an SVD of matrix $N_{bi}(k)H_{ds,i}(k)H_{ds,i}^T(k)N_{bi}^T(k)$.*

It is noted that the solution to the minimization problem (5) is not unique. For a solution $P_{si}^*(k) = \Sigma_{ki}^{-\frac{1}{2}} U_{ki}^T$ obtained in Proposition 2, $\Sigma_{ki}^{-\frac{1}{2}} U_{ki}^T N_{bi}(k) H_{fs,i}(k) H_{fs,i}^T(k) N_{bi}^T(k) U_{ki} \Sigma_{ki}^{-\frac{1}{2}}$ has an SVD given by

$$\Sigma_{ki}^{-\frac{1}{2}} U_{ki}^T N_{bi}(k) H_{fs,i}(k) H_{fs,i}^T(k) N_{bi}^T(k) U_{ki} \Sigma_{ki}^{-\frac{1}{2}} = U_{kf,i} \Sigma_{kf,i} U_{kf,i}^T, \; U_{kf,i}^T U_{kf,i} = I$$
$$\Sigma_p = \mathrm{diag}(\delta_{ki,1}, \delta_{ki,2}, \cdots, \delta_{ki,\gamma_{ki}}), \; \delta_{ki,1} \geq \delta_{ki,2} \geq \cdots \geq \delta_{ki,\gamma_{ki}} \geq 0, \; \delta_{ki,1} > 0$$

For any orthogonal matrix U_p with appropriate dimensions and diagonal matrix

$$\Sigma_p = \mathrm{diag}(\sigma_{p1}, \sigma_{p2}, \cdots, \sigma_{p\gamma_{ki}})$$

with $\sigma_{p1} \geq \sigma_{p2} \geq \cdots \geq \sigma_{p\gamma_{ki}} \geq 0$ and $\sigma_{p1} > 0$, it is easy to see that

$$\frac{\|U_p \Sigma_p^{-\frac{1}{2}} U_{kf,i}^T \Sigma_{ki}^{-\frac{1}{2}} U_{ki}^T N_{bi}(k) H_{ds,i}(k)\|}{\|U_p \Sigma_p^{-\frac{1}{2}} U_{kf,i}^T \Sigma_{ki}^{-\frac{1}{2}} U_{ki}^T N_{bi}(k) H_{fs,i}(k)\|} = \frac{\|\Sigma_p^{-\frac{1}{2}}\|}{\|\Sigma_p^{-\frac{1}{2}} \Sigma_{ki}^{-\frac{1}{2}}\|} = \frac{\sigma_{p1}}{\sigma_{p1}\delta_{ki,1}} = \frac{1}{\delta_{ki,1}}$$

from which we obtain the following result.

Proposition 3 *Let U_p be any orthogonal matrix with appropriate dimensions and*

$$\Sigma_p = \mathrm{diag}(\sigma_{p1}, \sigma_{p2}, \cdots, \sigma_{p\gamma_{ki}})$$

with $\sigma_{p1} \geq \sigma_{p2} \geq \cdots \geq \sigma_{p\gamma_{ki}} \geq 0$ and $\sigma_{p1} > 0$. Then, for any $\theta_k = i \in \Omega$, the set of all parity matrices satisfying the minimization problem (5) is parameterized by

$$W_{si}(k) = P_{si}(k) N_{bi}(k), \; P_{si}(k) = U_p \Sigma_p^{-\frac{1}{2}} U_{kf,i}^T \Sigma_{ki}^{-\frac{1}{2}} U_{ki}^T$$

Remark 3 *Proposition 3 implies that the solution to parity matrix $W_{si}(k)$ is not unique. One special choice is $W_{si}(k) = U_{kf,i}^T \Sigma_{ki}^{-\frac{1}{2}} U_{ki}^T N_{bi}(k)$.*

Remark 4 *In (5), the largest singular value of $P_{si}(k)N_{bi}(k)H_{ds,i}(k)$ and $P_{si}(k)N_{bi}(k)H_{fs,i}(k)$ are respectively used to represent the robustness of residual to unknown input and the sensitivity of residual to fault. The parity space-based FDI formulation using (5) means a best-case handling of the influence of faults on residual vector. In the case of $\Sigma_{ki}^{-\frac{1}{2}} U_{ki}^T N_{bi}(k) H_{fs,i}(k)$ being full-row rank, an alternative way is to use the smallest singular value of $P_{si}(k)N_{bi}(k)H_{fs,i}(k)$ as the sensitivity to fault, which is refereed to $\|P_{si}(k)N_{bi}(k)H_{fs,i}(k)\|_-$, and formulate the residual generation problem as*

$$\min_{P_{si}(k)} \frac{\|P_{si}(k)N_{bi}(k)H_{ds,i}(k)\|}{\|P_{si}(k)N_{bi}(k)H_{fs,i}(k)\|_-}, \; k = 0, 1, \cdots \tag{18}$$

Formulation (18) means a worst-case handling of the influence of faults on residual vector. More details about performance index of parity space-based approach, please refer to [4].

We are now in the position to implement on-line residual generation for system (1). We compute the N possible values of matrices $H_{os}(\theta_k),\ H_{ds}(\theta_k),\ H_{fs}(\theta_k),\ N_b(\theta_k)$, and $P_s(k)$ after time step $k-1$. When $y(k),\ u(k)$, and θ_k are available, one can calculate residual $r(\theta_k)$ using algebraic equation (4) immediately. The on-line implement of parity space-based residual generation is summarized into the following Algorithm 2.

Algorithm 2:

Step 1. Set $k_0 \geq s$. Denote by i_j the mode θ_j at $k = j$ for $j = k_0 - s, k_0 - s + 1, \cdots, k_0$.

Step 2. For $\theta_{k_0} = i_{k_0}$, calculate $H_{ds}(\theta_{k_0}),\ \Gamma(k_0),\ H_{us}(\theta_{k_0}),\ H_{fs}(\theta_{k_0})$, and $H_{os}(\theta_{k_0})$ using (9)–(11), (12)–(15) and (16)–(17), respectively, and let

$$H_{os,i_{k_0}}(k_0) = H_{os}(\theta_{k_0}),\ H_{us,i_{k_0}}(k_0) = H_{us}(\theta_{k_0}),\ H_{ds,i_{k_0}}(k_0) = H_{ds}(\theta_{k_0}),\ H_{fs,i_{k_0}}(k_0) = H_{fs}(\theta_{k_0})$$

Step 3. Let $k = k_0 + 1$. For all $\theta_k = i \in \Omega$, compute $H_{os}(\theta_k),\ H_{ds}(\theta_k)$, and $H_{fs}(\theta_k)$ using (9)–(11), (12)–(15) and (16)–(17) again and let

$$H_{os,i}(k) = H_{os}(\theta_k),\ H_{ds,i}(k) = H_{ds}(\theta_k),\ H_{fs,i}(k) = H_{fs}(\theta_k)$$

Step 4. Find a basis matrix $N_{bi}(k)$ of parity space and let

$$\gamma_i(k) = \mathrm{rank}(N_{bi}(k)),\ \beta_i(k) = \mathrm{rank}(N_{bi}(k)H_{ds,i}(k))$$

Step 5. If $\beta_i(k) < \gamma_i(k)$, then compute $P_{si}(k)$ by solving $P_{si}(k)N_{bi}(k)H_{ds,i}(k) = 0$ and $W_{si}(k) = P_{si}(k)N_{bi}(k)$; if $\beta_i(k) = \gamma_i(k)$, then we design $W_{si}(k)$ using Proposition 2 and 3.

Step 6. When $y(k),\ u(k)$ and θ_k are available, generate residual $r(\theta_k)$ using algebraic equation (4) immediately.

After residual generation, the remaining important task of fault detection is the residual evaluation. In this paper, we choose the 2−norm of $r_{i_k}(k)$ as a residual evaluation function $J_r(k)$. Noticing that

$$\|P_{si}(k)N_{bi}(k)H_{ds,i}(k)d_s(k)\| \leq \bar{\sigma}(P_{si}(k)N_{bi}(k)H_{ds,i}(k))\|d_s(k)\|$$

then the fault free case $J_r(k)$ becomes

$$\begin{cases} J_r(k) = 0, & \text{if } \beta_i(k) < \gamma_i(k) \\ J_r(k) \leq \sigma_{p1}\|d_s(k)\|, & \text{if } \beta_i(k) = \gamma_i(k) \end{cases}$$

Under the assumption of $d(k)$ being 2-norm bounded, there exists a constant $M > 0$ such that $\|d_s(k)\| \leq M$. Choosing a threshold J_{th} as

$$J_{th}(k) = \begin{cases} 0, & \text{if } \beta_i(k) < \gamma_i(k) \\ \sigma_{p1}M, & \text{if } \beta_i(k) = \gamma_i(k) \end{cases}$$

thus we can use (6) to detect the occurrence of a fault and deliver a fault alarm information.

Remark 5 *It is noted from Proposition 3 that the choice of Σ_p has no influence on performance index (5), while the selected $J_{th}(k)$ depends on σ_{p1}.*

4 A numerical example

To illustrate the proposed approach, we consider system (1) with $\theta_k \in \Omega = \{1,2,3\}$ and

$$
\begin{aligned}
A_1 &= \begin{bmatrix} -0.3 & 0 & 1 \\ 0 & 0.1 & 0.5 \\ 0 & -0.3 & -0.2 \end{bmatrix}, A_2 = \begin{bmatrix} -0.4 & 0.1 & 0 \\ 0.3 & 0.1 & -0.2 \\ 0.2 & 0 & 0.3 \end{bmatrix}, A_3 = \begin{bmatrix} 0.5 & 0 & 0.4 \\ -0.3 & -0.2 & 0.3 \\ 0 & 0.1 & 0.4 \end{bmatrix} \\
B &= \begin{bmatrix} 1 \\ 0.5 \\ 1 \end{bmatrix}, B_d = \begin{bmatrix} -0.8 \\ 0.6 \\ 0.5 \end{bmatrix}, B_f = \begin{bmatrix} 0.6 \\ 1 \\ 1 \end{bmatrix}, C_1 = \begin{bmatrix} 1 & 0 & 1 \end{bmatrix} \\
C_2 &= = \begin{bmatrix} 0 & 1 & 1 \end{bmatrix},\ C_3 \begin{bmatrix} 1 & 1 & 0 \end{bmatrix},\ D_d = 0.8,\ D_f = 0
\end{aligned}
$$

Set $s = 5,\ k_0 = 5$. For $k = 1, 2, \ldots, 100$, it is assumed that the mode θ_k changes as shown in Figure 1. Applying Algorithm 1, one can pre-design the finite points of parity matrix $W_s(\theta_k)$ at every step k on-line. For $k = 5$, we have

$$
W_{s1}(5) = \begin{bmatrix} -0.0098 & -0.0478 & -0.0622 & -0.6490 & -0.0100 & -0.2170 \\ 0.0057 & 0.0153 & 0.0601 & -0.1854 & -0.6679 & 0.5645 \\ -0.0043 & -0.0037 & 0.0360 & -0.3721 & 1.0002 & 1.0574 \end{bmatrix},
$$

$$
W_{s2}(5) = \begin{bmatrix} -0.0003 & -0.0097 & -0.0095 & -0.1955 & -0.3187 & -0.5184 \\ 0.0095 & 0.0437 & 0.0455 & 0.6820 & -0.0860 & -0.2061 \\ 0.0102 & 0.0272 & 0.0651 & 0.0719 & -1.4533 & 0.8647 \end{bmatrix},
$$

$$
W_{s3}(5) = \begin{bmatrix} -0.0131 & -0.0570 & -0.1038 & -0.4280 & 0.0806 & -0.4064 \\ 0.0039 & 0.0043 & 0.0572 & -0.5027 & -0.5188 & 0.4111 \\ -0.0010 & 0.0051 & 0.0637 & -0.5298 & 1.1259 & 0.7641 \end{bmatrix},
$$

The matrices $W_{si}(k)$ at $k = 6, 7, \cdots, 100$ are not listed here for simplicity.

Suppose that the unknown input $d(k)$ is a band-limited white noise as in Figure 2. Considering two different kinds of faults, i.e. (i) a pulse fault of unit amplitude occurred at $k = 40, 41, \cdots, 80$ (and is zero otherwise); and (ii) a sine wave fault with unit amplitude and angle frequency $\pi/20$ occurred at $k = 40, 41, \cdots, 80$ (and zero others). Figure 3 and Figure 4 show the two cases residual evaluation functions $J_r(k)$ (solid line) and thresholds $J_{th}(k)$ (dashed line) for $k = 5, 6, \ldots, 100$, respectively. The simulation results show that the fault alarm information can be delivered after two time steps and 6 steps of its occurrence, respectively.

5 Conclusion

The problem of parity space-based fault detection for a class of discrete-time linear Markovian jump systems has been investigated. To avoid repeated numerical operations, recursive formulas have been proposed firstly and, as a result, the computation burden of the parity equation coefficient matrices can be reduced. Then, a unified parity space-based approach can

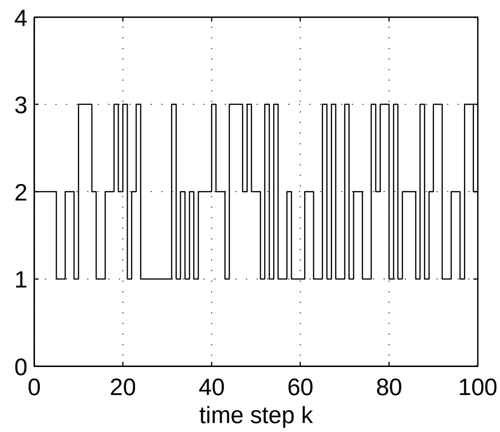

Figure 1: The mode θ_k

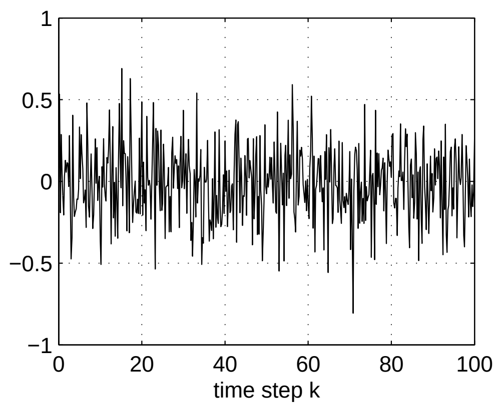

Figure 2: The unknown input $d(k)$

be extended to the fault detection for Markovian jump systems and an algorithm has been presented to implement the on-line residual generation. Furthermore, the 2-norm of residual signal has been chosen as a residual evaluation function and a corresponding threshold has been determined. Finally, a numerical example has been given to illustrate the effectiveness of the proposed method.

References

[1] J. Chen, R. J. Patton, *Robust model-based fault diagnosis for dynamic systems.* Boston: Kluwer Academic Publishers, 1999.

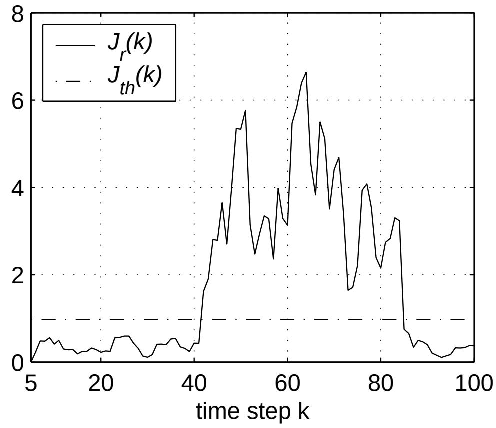

Figure 3: The residual evaluation function for Case 1

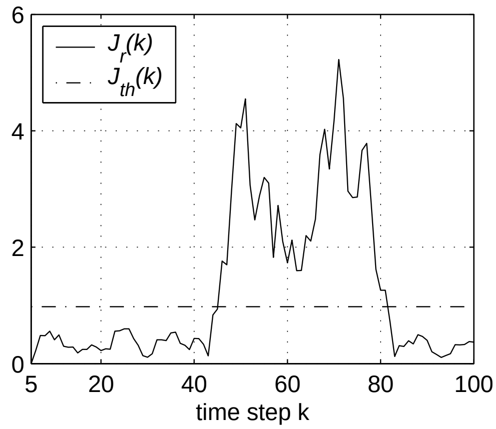

Figure 4: The residual evaluation function for Case 2

[2] R. J. Patton, J. Chen, "Parity space approach to model-based fault diagnosis", in *Proc. SAFEPROCESS'1994*, Baden-Baden, Germany, 1994, pp. 256–261.

[3] S. X. Ding, E. L. Ding, and T. Jeinsch, "An approach to analysis and design of observer and parity relation based FDI systems", in *Proceedings of the 14th World Congress of IFAC*, Beijing, China, July, 1999, pp. 37-41.

[4] S. X. Ding, E. L. Ding, T. Jeinsch, and P. Zhang, "An approach to a unified design of FDI system", in *Proc. of the 3rd Asia Control Conference*, Shanghai, China, July, 2000, pp. 2812-2817.

[5] P. M. Frank, S. X. Ding, and B. Koppen-Seliger, "Current developments in the theory of FDI," in *Proc. SAFEPROCESS'2000*, Budapest, Hungary, 2000, pp. 16–27.

[6] P. M. Frank, X. Ding, "Survey of robust residual generation and evaluation methods in observer-based fault detection systems," *Int. J. Process Control*, vol. 7, no. 6, pp.403–424,1997.

[7] A. Fekih, H. Xu, and F. N. Chowdhury, "Neural networks based system identification techniques for model based fault detection of nonlinear systems," *Int. J. Innovative Computing, Information and Contro*, vol. 3, no. 5, pp. 1053–1071, 2007.

[8] H. Gao, J. Lam, S. Xu, and C. Wang, "Stabilization and H_∞ control of two-dimensional markovian Jump systems," *IMA J. Mathematical Control and Information*, vol. 21, no. 4, pp. 377–392, 2004.

[9] J. Gertler, *Fault Detection and Diagnosis in Engineering Systems.* New York: Marcel Dekker, 1998.

[10] Z. Mao, B. Jiang, P. Shi, "$\mathcal{H}_\infty$ fault detection filter design for networked control systems modelled by discrete Markovian jump systems," *IET Proc. Control Theory and Applications*, vol. 1, no. 5, pp. 1336–1343, 2007.

[11] Z. Mao, B. Jiang, "Fault identification and fault-tolerant control for a class of networked control systems," *Int. J. Innovative Computing, Information and Contro*, vol. 3, no. 5, pp. 1121–1130, 2007.

[12] S. K. Nguang, P. Shi and S.X. Ding, "Fault detection for uncertain fuzzy systems: an LMI approach," *IEEE Trans. Fuzzy Systems*, vol. 15, no. 6, pp. 1251-1262, 2007.

[13] Z. Wang, J. Lam, X. Liu, "Robust filtering for discrete-time Markovian jump delay systems," *IEEE Signal Processing Letters*, vol. 11, no. 8, pp. 659–662, 2004.

[14] H. Wang, C. Wang, H. Gao, and L. Wu, "An LMI approach to fault detection and isolation filter design for Markovian jump system with mode-dependent time-delays," in *Proc. of the 2006 American Control Conference* Minnesota, USA, 2006, pp. 5686–5691.

[15] L. Wu, P. Shi, H. Gao, and C. Wang, "H_∞ filtering for 2D Markovian jump systems," *Automatica*, (2008), doi:10.1016/j.automatica.2007.10.

[16] M. Zhong, S. X. Ding, J. Lam, and H. Wang, "LMI approach to design robust fault detection filter for uncertain LTI systems," *Automatica*, vol. 39, no. 3, pp. 543–550, 2003.

[17] M. Zhong, J. Lam, S. X. Ding, and P. Shi, "Robust fault detection of Markovian jump systems," *Circuits Systems Signal Processing*, vol. 23, no. 5, pp. 307–407, 2004.

[18] M. Zhong, H. Ye, P. Shi, and G. Wang, "Fault detection for Markovian jump systems," *IEE Proc. Control Theory and Applications*, vol. 152, no. 4, pp. 397–402, 2005.

[19] K. Zhou, J. C. Doyle, *Essentials of Robust Control.* New Jersey: Prentice Hall, 1998.