2008 American Control Conference
Westin Seattle Hotel, Seattle, Washington, USA
June 11-13, 2008

Parity Space Fault Detection Based on Irregularly Sampled Data

Iman Izadi, Sirish L. Shah and Tongwen Chen

Abstract— **In this paper, we study fault detection and isolation in non-uniformly sampled systems. In these systems the control input is generated and the process output is sampled at non-uniformly distributed time instants. A parity space residual generator is constructed and optimized to minimize the sensitivity of the residual to disturbances. The residual generator is developed assuming that fault and disturbance signals are constant during the sampling intervals. This assumption is practically acceptable if sampling intervals are small. For larger sampling intervals, a method is proposed to improve the performance. No periodicity assumption is made for the sampling instants (i.e., irregularly sampled data is considered).**

I. Introduction

Industrial control systems consist of many components, including sensors, actuators, controllers, communication networks and computer hardware/software. Any abnormality, deficiency or malfunction in one of these components can disrupt the normal operation of the system and lead to performance degradation, instability, failure (total breakdown of the system) and even dangerous situations. To maintain an acceptable level of quality, cost efficiency, reliability and safety, it is important that abnormal behavior of a system component, usually referred to as a fault, be promptly detected and appropriate remedies be applied. Fault detection has been a very active area of research both in industry and academia in the past decades. A wide variety of fault detection and isolation (FDI) methods are available in the literature based on various control, mathematical and statistical concepts [1], [2].

A control system in which the control and FDI algorithms are digitally implemented on a computer is known as a sampled-data control system. In such systems, the actual process under control which is often a continuous-time process, is connected to the computer network through analog-to-digital (A/D) and digital-to-analog (D/A) converters. Conventionally, it is assumed that each process variable is sampled at a constant rate and each control signal is also generated at a constant rate. The sampling rates of different A/D and D/A converters may be equal and synchronous (i.e., single-rate systems) or different and/or asynchronous (i.e., multirate systems). However, this is not the case in many practical situations, for instance in chemical processes. Frequently, the process inputs and outputs are generated and sampled at non-uniformly spaced time instants. This could happen due to a number of reasons, including unpredictable delays in sensors, communication network and laboratory analysis. Also in task-sharing applications, where the control algorithm is implemented on the same distributed computer system that monitors the process and manages other aspects of the plant, it is more reasonable and cost-effective to allow non-uniform sampling. Moreover, it has been shown that non-uniformly sampled systems can have some advantages over uniformly sampled ones [3], [4].

In this paper, we develop an optimal residual generator for non-uniformly sampled systems based on the parity space approach. The development consists of three main steps: obtaining the relationship between the input and output of the process in a certain time frame; construction of the residual generator which utilizes some design parameter; and optimal selection of the design parameter. In previous works on non-uniformly sampled systems, it is assumed that sampling, although non-uniform, follows a periodic pattern. In other words, the sampling and updating instants are non-uniformly distributed in a window of time, and this window is periodically repeated. This allows the use of lifting technique to obtain a linear time-invariant model of the process. But the periodicity assumption is too restrictive as not many non-uniformly sampled systems follow this pattern. In this paper, we don't make any periodicity assumptions and the sampling and updating instants can be arbitrarily distributed over time. Therefore, the proposed residual generator is applicable to general non-uniformly sampled systems. Due to this non-periodicity assumption, the lifting technique can not be used. Instead, we use a time-varying formulation to approach non-uniformly sampled systems, which is basically different from the lifting approach used in previous works.

In the proposed method, we also assume that the fault and disturbance inputs are constant over the sampling interval (i.e., indirect design [5], [6]). This assumption is practically acceptable only if the sampling intervals are sufficiently small. If not, a technique is introduced to improve the performance. Another approach is to use the so called-direct design [7], [6]. In direct design no assumption is made on the fault and disturbance signals and they can vary freely over time. As a result, operators should be used to capture the effect of continuous-time fault and disturbance on discrete-time residual. A direct approach for fault detection in non-uniformly sampled systems has also been developed by the authors [8].

A number of research results is available on control [9],

This research was supported by NSERC-Matrikon-Suncor-iCORE senior industrial research chair program at the University of Alberta.

I. Izadi and S.L. Shah are with the Department of Chemical & Materials Engineering, University of Alberta, Edmonton, Alberta, Canada, T6G 2G6, iman@ualberta.ca, sirish.shah@ualberta.ca

T. Chen is with the Department of Electrical & Computer Engineering, University of Alberta, Edmonton, Alberta, Canada, T6G 2V4, tchen@ece.ualberta.ca

978-1-4244-2079-7/08/$25.00 ©2008 AACC.

[4], and fault detection of non-uniformly sampled systems [10], [11]. In [10], a subspace approach was proposed to identify residual models which were used for fault detection. In [11] a Kalman filter based FDI was developed. As mentioned before, in all of these works, it was assumed that the sampling/updating is non-uniform but periodic, and then the lifted model was derived. In this paper no such assumption is made.

II. Preliminaries

A. Parity Space Approach [12]

Consider the following discrete-time system

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) + Ed(k) + Ff(k) \\ y(k) = Cx(k) \end{cases}$$

where $x(k) \in \mathbb{R}^{n_x}$ is the state vector, $u(k) \in \mathbb{R}^{n_u}$ the vector of control input, $y(k) \in \mathbb{R}^{n_y}$ the vector of process output, $d(k) \in \mathbb{R}^{n_d}$ the vector of unknown inputs (e.g., disturbance, noise, model mismatch, etc.) and $f(k) \in \mathbb{R}^{n_f}$ the vector of faults to be detected. A, B, C, E, and F are known matrices of appropriate dimensions. The objective of residual generation is to use the known variables of the process (i.e., current and past values of $y(k)$ and $u(k)$) to generate a fault-indicating signal known as residual. The fault is then detected by evaluating the residual, which usually is a simple check to see if the threshold is exceeded.

For a fixed number s, known as the order of parity relation, define

$$y_s(k) = \begin{bmatrix} y(k-s) \\ y(k-s+1) \\ \vdots \\ y(k) \end{bmatrix}_{(s+1)n_y \times 1}.$$

$u_s(k)$, $d_s(k)$ and $f_s(k)$ are also defined similarly. It is straight forward to show that $y_s(k)$, $u_s(k)$, $d_s(k)$ and $f_s(k)$ are related through the following expression

$$y_s(k) = H_o x(k-s) + H_u u_s(k) + H_d d_s(k) + H_f f_s(k), \quad (1)$$

where

$$H_o = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^s \end{bmatrix}, H_u = \begin{bmatrix} 0 & \cdots & 0 & 0 \\ CB & \cdots & 0 & 0 \\ \vdots & & \vdots & \vdots \\ CA^{s-1}B & \cdots & CB & 0 \end{bmatrix}.$$

H_d and H_f are similar to H_u with B replaced by E and F respectively. Based on (1), a parity space based residual generator is formulated as

$$r(k) = v_s\big(y_s(k) - H_u u_s(k)\big),$$

where $r(k) \in \mathbb{R}$ is the residual. The design parameter $v_s \in \mathbb{R}^{1\times(s+1)n_y}$, known as the parity vector, is chosen from the parity space P_s defined by

$$P_s = \{v_s | v_s H_o = 0\}.$$

Substituting $y_s(k)$ from (1), the dynamics of the residual generator is expressed by

$$r(k) = v_s\big(H_d d_s(k) + H_f f_s(k)\big), \quad v_s \in P_s.$$

If the residual $r(k)$ can not be perfectly decoupled from the unknown input $d(k)$, the effect of $d(k)$ on $r(k)$ can be minimized by optimizing the residual generator. A common choice of performance index for optimization is

$$J = \frac{\|v_s H_d\|^2}{\|v_s H_f\|^2} = \frac{v_s H_d H_d^T v_s^T}{v_s H_f H_f^T v_s^T}.$$

The numerator of J reflects the effect of unknown input $d(k)$ on the residual while the denominator reflects the effect of fault $f(k)$. By minimizing J a compromise is made between sensitivity to fault and robustness to disturbances. For the solution of the optimization problem, assume that N_B is the basis vector for parity space P_s. Also assume that $\lambda_{\min}$ is the minimum generalized eigenvalue of the pair $\left(N_B H_d H_d^T N_B^T, N_B H_f H_f^T N_B^T\right)$ and $p_{s,\min}$ is the corresponding generalized eigenvector, i.e.,

$$p_{s,\min} N_B H_d H_d^T N_B^T = \lambda_{\min} p_{s,\min} N_B H_f H_f^T N_B^T.$$

Then $v_s^* = p_{s,\min} N_B$ is the optimal solution and $J^* = \lambda_{\min}$ is the optimal performance.

B. Process Description

Consider an LTI, strictly proper, continuous-time system controlled by a discrete-time controller in a sampled-data framework. The process under control is driven by three sets of inputs: a computer generated control input; an unknown input to represent disturbance, noise, model mismatch and other uncertainties; and fault input. The state-space representation of the continuous-time process is given by

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) + Ed(t) + Ff(t) \\ y(t) = Cx(t) \end{cases} \quad (2)$$

where $x(t) \in \mathbb{R}^{n_x}$ is the state vector, $u(t) \in \mathbb{R}^{n_u}$ the known vector of control input, $y(t) \in \mathbb{R}^{n_y}$ the vector of process output, $d(t) \in \mathbb{R}^{n_d}$ the vector of unknown input and $f(t) \in \mathbb{R}^{n_f}$ the vector of fault to be detected. A, B, C, E and F are known matrices of appropriate dimensions. The assumption of strict properness is standard in the sampled-data literature and necessary for boundedness of the sampling operator. In practice, because of antialiasing filters that are used before sampling, the systems are always strictly proper. Also notice that due to antialiasing filters, $f(t)$ can represent sensor as well as actuator and process faults.

The control signal $u(t)$ is generated by the computer at non-uniformly spaced time instants and the output is also sampled at non-uniformly spaced time instants (hence the term non-uniformly sampled system). For simplicity, here we assume that the inputs/outputs are generated/sampled synchronously at the same times. Without this assumption, the derivation of the residual generator is more complex but quite straight forward.

Let $T = \{t_0, t_1, t_2, \cdots\}$ be the set of time instants when the output is sampled (or the input is updated). Let $\ell_T(\mathbb{Z})$ denote the vector space of all discrete-time signals corresponding to the time instants in T. Notice that the discrete-time signals in $\ell_T(\mathbb{Z})$ has no practical meaning unless the corresponding time instants, given by T, are known. Let $\mathcal{L}(\mathbb{R})$ denote the vector space of all continuous-time signals.

The non-uniform D/A converter is modeled by non-uniform (zero-order) hold operator $H_T : \ell_T(\mathbb{Z}) \rightarrow \mathcal{L}(\mathbb{R})$ defined as

$$u(t) = H_T \upsilon_T(k) = \upsilon_T(k), \quad t_k \leq t < t_{k+1}.$$

Here $\upsilon_T(k)$ represent the discrete-time input. Notice that since $u(t)$ is the output of a hold operator, it is piecewise constant.

The non-uniform A/D converter is also modeled by non-uniform sampling operator $S_T : \mathcal{L}(\mathbb{R}) \rightarrow \ell_T(\mathbb{Z})$ defined as

$$\psi_T(k) = S_T y(t) = y(t_k),$$

where $\psi_T(k)$ denotes the discrete-time output.

In general, disturbance and fault signals, $d(t)$ and $f(t)$, can have arbitrary values at any time. However, for the method discussed in this paper (known as the indirect design [5], [6]), we assume that $d(t)$ and $f(t)$ are piecewise constant according to T. In other words

$$\begin{aligned} d(t) &= H_T S_T d(t), \\ f(t) &= H_T S_T f(t). \end{aligned}$$

This assumption is obviously restrictive, but can be practically acceptable if the sampling intervals are small. A technique will be introduced later in Section IV to improve the performance if the sampling intervals are large. Define

$$\begin{aligned} \delta_T(k) &= S_T d(t), \\ \phi_T(k) &= S_T f(t). \end{aligned}$$

III. Indirect Method of Residual Generation

The core of the parity space approach, presented in Section II for discrete-time systems, is eqn. (1). This equation shows how the output of the system within an interval of time ($(s+1)h$ sec here, where h is the sampling period) is affected by the state of the system at the beginning of the interval and the inputs of the system (including controlled input, disturbance and fault) during the interval. The first step in constructing a residual generator for non-uniformly sampled systems is to derive an expressions similar to (1). To do that, at each sampling instant t_k, we select a time frame that contains $s+1$ samples of the output $\big(\psi_T(k-s)$ to $\psi_T(k)\big)$. Therefore, the time frame is $[t_{k-s}, t_k]$. Notice that due to the non-uniform sampling pattern, the actual length of the time frame is different at each time instant. Define

$$\psi_{T,s}(k) = \begin{bmatrix} \psi_T(k-s) \\ \psi_T(k-s+1) \\ \vdots \\ \psi_T(k) \end{bmatrix}_{(s+1)n_y \times 1}.$$

$\upsilon_{T,s}(k)$, $\delta_{T,s}(k)$ and $\phi_{T,s}(k)$ are defined similarly. Here the objective is to express $\psi_{T,s}(k)$ in terms of the state of the system at the beginning of the time frame $\big(x(t_{k-s})\big)$ and $\upsilon_{T,s}(k)$, $\delta_{T,s}(k)$ and $\phi_{T,s}(k)$.

For now assume that $d(t) = 0$ and $f(t) = 0$. It is well known that, for any two times $\tau_1 \leq \tau_2$,

$$x(\tau_2) = e^{(\tau_2-\tau_1)A} x(\tau_1) + \int_{\tau_1}^{\tau_2} e^{(\tau_2-\tau)A} B u(\tau) d\tau.$$

By substituting $\tau_1 = t_{k-s}$ and $\tau_2 = t_{k-s+i}, \ i = 0, 1, \cdots s$ in this equation we get

$$\begin{aligned} x(t_{k-s+i}) &= e^{(t_{k-s+i}-t_{k-s})A} x(t_{k-s}) \\ &\quad + \int_{t_{k-s}}^{t_{k-s+i}} e^{(t_{k-s+i}-\tau)A} B u(\tau) d\tau \\ &= e^{(t_{k-s+i}-t_{k-s})A} x(t_{k-s}) \\ &\quad + \sum_{m=1}^{i} \int_{t_{k-s+m-1}}^{t_{k-s+m}} e^{(t_{k-s+i}-\tau)A} B u(\tau) d\tau. \end{aligned} \tag{3}$$

But in the interval $[t_{k-s+m-1}, t_{k-s+m})$, the input is constant: $u(t) = \upsilon_T(k-s+m-1)$. Therefore, we can rewrite the last term of the above equation as

$$\begin{aligned} &\int_{t_{k-s+m-1}}^{t_{k-s+m}} e^{(t_{k-s+i}-\tau)A} B u(\tau) d\tau \\ &\quad = \int_{t_{k-s+m-1}}^{t_{k-s+m}} e^{(t_{k-s+i}-\tau)A} d\tau B \upsilon_T(k-s+m-1) \\ &\quad = e^{(t_{k-s+i}-t_{k-s+m})A} \int_{t_{k-s+m-1}}^{t_{k-s+m}} e^{(t_{k-s+m}-\tau)A} d\tau B \\ &\qquad\qquad \times \upsilon_T(k-s+m-1). \end{aligned}$$

Now for $\tau_1 \leq \tau_2$ define

$$\begin{aligned} A_d(\tau_1, \tau_2) &= e^{(\tau_2-\tau_1)A}, \\ B_d(\tau_1, \tau_2) &= \int_{\tau_1}^{\tau_2} e^{(\tau_2-\tau)A} d\tau B = \int_0^{\tau_2-\tau_1} e^{\tau A} d\tau B. \end{aligned}$$

Then (3) can be simplified as

$$\begin{aligned} x(t_{k-s+i}) &= A_d(t_{k-s}, t_{k-s+i}) x(t_{k-s}) \\ &\quad + \sum_{m=1}^{i} A_d(t_{k-s+m}, t_{k-s+i}) B_d(t_{k-s+m-1}, t_{k-s+m}) \\ &\qquad\qquad \times \upsilon_T(k-s+m-1) \end{aligned}$$

The output equation in (2) implies that $\psi_T(k) = y(t_k) = Cx(t_k)$ and then

$$\begin{aligned} \psi_T(k-s+i) &= CA_d(t_{k-s}, t_{k-s+i}) x(t_{k-s}) \\ &\quad + \sum_{m=1}^{i} CA_d(t_{k-s+m}, t_{k-s+i}) B_d(t_{k-s+m-1}, t_{k-s+m}) \\ &\qquad\qquad \times \upsilon_T(k-s+m-1) \end{aligned}$$

By changing i from 0 to s, and stacking all the equations we get

$$\psi_{T,s}(k) = H_{o,T}(k) x(t_{k-s}) + H_T(k) H_{B_d,T}(k) \upsilon_{T,s}(k),$$

where $H_{o,T}(k) : (s+1)n_y \times n_x$, $H_T(k) : (s+1)n_y \times (s+1)n_x$ and $H_{B_d,T}(k) : (s+1)n_x \times (s+1)n_u$ are given by

$$H_{o,T}(k) = \begin{bmatrix} C \\ CA_d(t_{k-s}, t_{k-s+1}) \\ CA_d(t_{k-s}, t_{k-s+2}) \\ \vdots \\ CA_d(t_{k-s}, t_k) \end{bmatrix},$$

$$H_T(k) = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ C & 0 & \cdots & 0 & 0 \\ CA_d(t_{k-s+1}, t_{k-s+2}) & C & \cdots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ CA_d(t_{k-s+1}, t_k) & CA_d(t_{k-s+2}, t_k) & \cdots & C & 0 \end{bmatrix},$$

$$H_{B_d,T}(k) = \begin{bmatrix} B_d(t_{k-s}, t_{k-s+1}) & \cdots & 0 \\ \vdots & & \vdots \\ 0 & \cdots & B_d(t_k, t_{k+1}) \end{bmatrix}. \quad (4)$$

Now for the general case when $d(t) \neq 0$ and $f(t) \neq 0$, we have

$$\begin{aligned} \psi_{T,s}(k) = H_{o,T}(k)x(t_{k-s}) &+ H_T(k)H_{B_d,T}(k)\upsilon_{T,s}(k) \\ &+ H_T(k)H_{E_d,T}(k)\delta_{T,s}(k) \\ &+ H_T(k)H_{F_d,T}(k)\phi_{T,s}(k). \end{aligned} \quad (5)$$

Here $H_{E_d,T}(k)$ and $H_{F_d,T}(k)$ have the same structure as $H_{B_d,T}(k)$ in (4) with B replaced by E and F respectively. This equation shows the relationship between the non-uniformly sampled output of the system within the interval $[t_{k-s}, t_k]$ with the state of the system at the beginning of the interval and the known and unknown inputs of the system within the interval.

Based on (5), a parity space based residual generator for the non-uniformly sampled system is formulated as

$$r(k) = v_s(k)\big(\psi_{T,s}(k) - H_T(k)H_{B_d,T}(k)\upsilon_{T,s}(k)\big). \quad (6)$$

Here, $r(k) \in \mathbb{R}$ is the residual and s is the order of parity relation. The parity vector $v_s(k) \in \mathbb{R}^{1\times(s+1)n_y}$ is the design parameter. Since the non-uniformly sampled system described above is inherently time-varying, the residual generator should also be time-varying. This is why the parity vector is a function of k and should be calculated at each iteration. The parity vector $v_s(k)$ belongs to the parity space $P_s(k)$ given by

$$P_s(k) = \{v_s(k) | v_s(k)H_{o,T}(k) = 0\}.$$

Dynamics of the residual generator with respect to the discrete-time inputs $\delta_T(k)$ and $\phi_T(k)$ is then expressed by

$$\begin{aligned} r(k) = v_s(k)H_T(k)\Big(&H_{E_d,T}(k)\delta_{T,s}(k) \\ &+ H_{F_d,T}(k)\phi_{T,s}(k)\Big). \end{aligned}$$

If the parity vector $v_s(k)$ can be found such that

$$\begin{aligned} v_s(k)H_T(k)H_{E_d,T}(k) &\equiv 0, \\ v_s(k)H_T(k)H_{F_d,T}(k) &\neq 0, \end{aligned}$$

then the unknown input $\delta_T(k)$ has no effect on the residual and perfect disturbance decoupling is achieved. However, the conditions for perfect disturbance decoupling are very hard to satisfy and therefore an optimization procedure is considered. Similar to the discrete-time LTI case, $v_s(k)$ is obtained by optimizing the following performance index

$$J_d(k) = \frac{\|v_s(k)H_T(k)H_{E_d,T}(k)\|^2}{\|v_s(k)H_T(k)H_{F_d,T}(k)\|^2}$$

IV. Indirect Design with Interval Division

In the previous section, we assumed that the unknown input and fault signals are piecewise constant according to T, i.e., $d(t)$ and $f(t)$ are constant in the interval $[t_k, t_{k+1}),\ k = 0, 1, \cdots$. If the sampling intervals are relatively large, this assumption may result in performance degradation. In that case, we assume that $d(t)$ and $f(t)$ are constant over smaller intervals. Divide $[t_k, t_{k+1}),\ k = 0, 1, \cdots$ into l equal subintervals. Let T_l denote the set of all time instants obtained by this division, i.e.,

$$\begin{aligned} T_l &= \Big\{t_0, \tfrac{(l-1)t_0+t_1}{l}, \cdots, \tfrac{t_0+(l-1)t_1}{l}, t_1, \tfrac{(l-1)t_1+t_2}{l}, \cdots\Big\} \\ &= \Big\{t_{0,0}, t_{0,1}, \cdots, t_{0,l-1}, t_{1,0}, t_{1,1}, \cdots\Big\} \end{aligned}$$

Notice that $t_{k,0} = t_k$ and define $t_{k,l} = t_{k+1}$. Assume that $d(t)$ and $f(t)$ are piecewise constant functions of time according to T_l, i.e.,

$$\begin{aligned} d(t) &= H_{T_l}S_{T_l}d(t), \\ f(t) &= H_{T_l}S_{T_l}f(t). \end{aligned}$$

Then define

$$\begin{aligned} \delta_{T_l}(k) &= S_{T_l}d(t), \\ \phi_{T_l}(k) &= S_{T_l}f(t). \end{aligned}$$

Also define

$$\delta_{T_l,s}(k) = \begin{bmatrix} \delta_{T_l}(l(k-s)) \\ \vdots \\ \delta_{T_l}(l(k-s)+l-1) \\ \hdashline \delta_{T_l}(l(k-s+1)) \\ \vdots \\ \delta_{T_l}(l(k-s+1)+l-1) \\ \hdashline \vdots \\ \hdashline \delta_{T_l}(lk) \\ \vdots \\ \delta_{T_l}(lk+l-1) \end{bmatrix}_{(s+1)ln_d\times 1},$$

and $\phi_{T_l,s}(k)$ similarly. The number of subintervals l, can be selected based on the length of the sampling intervals and the expected behavior of the disturbance and fault inputs.

Similar to the discussion in Section III, it can be shown that the relation between the input $\big(\upsilon_{T,s}(k),\ \delta_{T_l,s}(k),\ \phi_{T_l,s}(k)\big)$ and output $\big(\psi_{T,s}(k)\big)$ in time frame $[t_{k-s}, t_k]$ is given by

$$\begin{aligned}\psi_{T,s}(k) = H_{o,T}(k)x(t_{k-s}) & \\ + H_T(k)H_{B_d,T}(k)\upsilon_{T,s}(k) & \\ + H_T(k)H_{E_\gamma,T}(k)\delta_{T_l,s}(k) & \\ + H_T(k)H_{F_\gamma,T}(k)\phi_{T_l,s}(k). &\end{aligned} \tag{7}$$

Here $H_{E_\gamma,T}(k) : (s+1)n_x \times (s+1)ln_d$ is defined as

$$H_{E_\gamma,T}(k) = \begin{bmatrix} E_\gamma(t_{k-s}, t_{k-s+1}) & \cdots & 0 \\ \vdots & & \vdots \\ 0 & \cdots & E_\gamma(t_k, t_{k+1}) \end{bmatrix},$$

where $E_\gamma(\tau_1, \tau_2) : n_x \times ln_d$ is

$$E_\gamma(\tau_1, \tau_2) = \begin{bmatrix} \int_{\tau_{1,0}}^{\tau_{1,1}} & \int_{\tau_{1,1}}^{\tau_{1,2}} & \cdots & \int_{\tau_{1,l-1}}^{\tau_{1,l}} \end{bmatrix} e^{(\tau_2-\tau)A} d\tau E,$$

and $\tau_{1,j} = \tau_1 + j\frac{\tau_2-\tau_1}{l},\ j = 0, \cdots, l$. Notice the similarity between $H_{E_\gamma,T}(k)$ and $H_{E_d,T}(k)$. Similarly define $F_\gamma(\tau_1, \tau_2)$ and $H_{F_\gamma,T}(k)$.

Based on the input-output relation in (7), a residual generator can be constructed which is the same residual generator given in (6). Consequently, the dynamics of the residual generator with respect to the discrete-time inputs $\delta_{T_l,s}(k)$ and $\phi_{T_l,s}(k)$ is given by

$$r(k) = v_s(k)H_T(k)\Big(H_{E_\gamma,T}(k)\delta_{T_l,s}(k) + H_{F_\gamma,T}(k)\phi_{T_l,s}(k)\Big).$$

If perfect decoupling of the residual from disturbance is not possible, the parity vector $v_s(k) \in P_s(k)$ is designed by minimizing the following performance index

$$J_\gamma(k) = \frac{\big\|v_s(k)H_T(k)H_{E_\gamma,T}(k)\big\|^2}{\big\|v_s(k)H_T(k)H_{F_\gamma,T}(k)\big\|^2}$$

V. Optimal Parity Vector

The indirect method of residual generation and its modification with interval division (presented in Sections III and IV), both resulted in the following performance index

$$J_x(k) = \frac{\|v_s(k)H_T(k)H_{E_x,T}(k)\|^2}{\|v_s(k)H_T(k)H_{F_x,T}(k)\|^2}, \tag{8}$$

where $x \equiv d$ for the indirect design and $x \equiv \gamma$ for the indirect design with interval division. It is important to keep in mind that regardless of the design method, we always use the residual generator in (6) for implementation. The techniques proposed in Sections III and IV are different approaches to design the parity vector $v_s(k)$, not different methods of implementation.

The parity vector $v_s(k)$ is now designed by solving the following optimization problem for $k = s, s+1, \cdots$,

$$\min_{v_s(k)\in P_s(k)} J_x(k).$$

Assume that $N_B(k)$ is the basis vector for parity space $P_s(k)$, and $\lambda_{\min}(k)$ and $p_{s,\min}(k)$ are the minimum generalized eigenvalue and the corresponding generalized eigenvector satisfying

$$p_{s,\min}(k)N_B(k)H_T(k)\Big(H_{E_x,T}(k)H_{E_x,T}^T(k) - \lambda_{\min}(k)H_{F_x,T}(k)H_{F_x,T}^T(k)\Big)H_T^T(k)N_B^T(k) = 0.$$

Then $v_s^*(k) = p_{s,\min}(k)N_B(k)$ is the optimal parity vector and $J^*(k) = \lambda_{\min}(k)$ is the optimal performance. Once the optimal parity vector $v_s^*(k)$ is designed, the residual in (6) can be implemented. Notice that this residual generator, updates the residual at time instants $t_k,\ k = s, s+1, \cdots$. These are the instants of time that the output is sampled and are in fact the best times to generate the residual. The reason is that at these times, new information from the process is available through measurements and therefore the fault can be detected at the earliest time possible.

After the residual is generated, it has to be evaluated (usually by comparing to a threshold) before a decision about fault occurrence can be made. It has been shown that the threshold depends on the optimal parity vector [1]. Therefore, since the optimal parity vector $v_s^*(k)$ is time-varying, the threshold will also be time-varying which can be unsatisfactory for implementation and visualization. Notice, on the other hand, that the parity vector $v_s^*(k)$ appears both in numerator and denominator of the performance index in (8). So, multiplying $v_s^*(k)$ by a scalar does not change the optimal performance $J^*(k)$. This scalar multiplication, however, affects the threshold. Therefore, the scalar multiplier can be used to normalize the threshold at each sampling instant resulting in a threshold that remains constant during the course of implementation.

As mentioned before, the residual generator designed for the non-uniformly sampled system is time-varying. Therefore, the related matrices $H_{o,T}(k)$, $H_T(k)$, $H_{B_d,T}(k)$, $H_{E_x,T}(k)$ and $H_{F_x,T}(k)$ should be recalculated and the optimization problem re-solved every time (the calculations at each step, mainly simple matrix computations and an eigenvector problem, are not numerically complex). In fact, any residual generator designed for the general non-uniformly sampled system is intuitively time-varying. This is because the non-uniformly sampled system is inherently time-varying and unpredictable (i.e., we do not know when to expect the next sample). However, if the non-uniform sampling follows a certain pattern that makes it predictable, then the matrices can be calculated before hand and the parity vector computed off-line. This is the case in other results for non-uniformly sampled systems in which a periodic sampling pattern is considered [9], [10], [11], [4].

VI. Example

Consider the LTI continuous-time process in (2) with

$$A = \begin{bmatrix} -1 & 5 \\ 0 & -2 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

$$E = \begin{bmatrix} 0.1 \\ 1 \end{bmatrix}, \quad F = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

The outputs are non-uniformly sampled at the time instants given by (in seconds)

$$T = \{0, 1.2, 1.6, 2.1, 3.0, 4.2, 4.9, 5.8, 7, 7.7, 8.5, 9.4, \\ 10.4, 11.1, 11.9, 13.0, 14.5, 15.4, 16.5, 17.1, 17.9\}.$$

The control input is also updated according to T with random numbers between -6 and 6. The disturbance $d(t)$ is white noise with variance 1 and is updated every 0.1 sec. The fault $f(t)$ is a step function, changing from 0 to 1 at 9.7 sec. The input and outputs of the system are shown in Fig. 1.

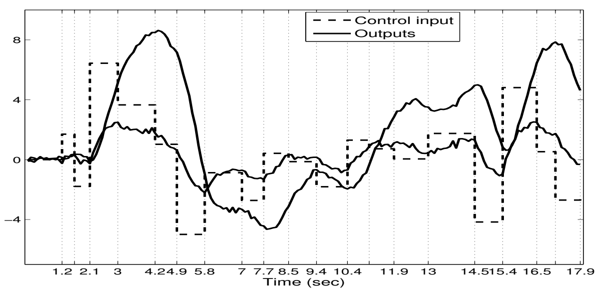

Fig. 1. System input and outputs

At 10.4 sec, the outputs are sampled for the first time after the fault occurrence, and this is the first time that fault information is available to the detection algorithm. Therefore, we expect to see the effect of the fault on the residual at 10.4 sec. For the order of parity relations we chose $s = 3$. Two optimal residual generators were designed for this non-uniformly sampled system. The first residual generator was designed using the indirect method. For the second one each sampling interval was divided into 3 sub-intervals (i.e., $l = 3$), and the residual generator was designed using the indirect design with interval division. For both designs, the threshold is set to be at 1. The results of simulation are shown in Fig. 2.

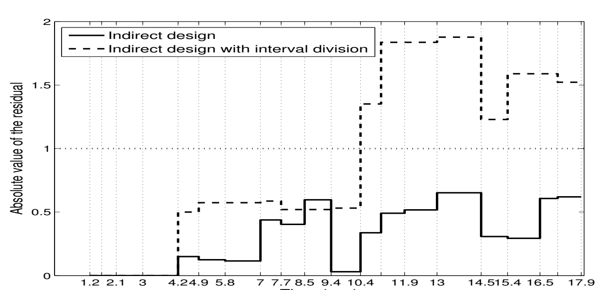

Fig. 2. Residual signal for fault detection

As it can be seen, the residual generated by the indirect design was not able to reflect the fault. But with performance improvement using interval division, the residual generator was able to detect the fault at the earliest possible time (10.4 sec).

VII. Conclusions

In this paper, we presented an indirect method to design optimal residual generator for non-uniformly sampled systems. The proposed parity based residual generator uses process input-output data in a certain frame of time to generate the residual. A key point in developing the residual generator is not to fix the length of the time frame, but to fix the number of data samples in that frame. The residual is then evaluated and a decision is made on fault occurrence. To design the optimal parity vector, it was assumed that the fault and disturbance inputs are constant during the sampling period (hence the indirect design). If the sampling intervals are small, we expect this assumption to be acceptable. For larger sampling intervals, to improve the performance of indirect design, each interval is divided into a selectable number of subinterval. Then it is assumed that the fault and disturbance inputs are constant over the subintervals.

Throughout the paper no assumption is made on the sampling and hold operators. Particularly, there is no need for the sampling and hold operators to follow a periodic pattern. In fact no *a priori* information is required regarding the sampling/updating times. Whenever a new measurement from the process becomes available, the residual can be updated. This makes the method applicable to general non-uniformly sampled systems and also eliminates any unnecessary delay in fault detection. Also, since multirate sampled-data systems are a special case of general non-uniformly sampled systems, the design techniques proposed in this paper can be readily used for multirate systems.

References

[1] J. Chen and R. Patton, *Robust Model-Based Fault Diagnosis for Dynamic Systems*, Kluwer Academic Publishers, Boston, 1999.

[2] R. Isermann, *Fault-Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance*, Springer, Berlin, 2006.

[3] G. Kreisselmeier, "On sampling without loss of observability controllability", *IEEE Trans. on Automatic Control*, vol. 44, pp. 1021-1025, 1999.

[4] J. Sheng, T. Chen and S.L. Shah, "Generalized predictive control for non-uniformly sampled systems", *Journal of Process Control*, vol. 12, pp. 875–885, 2002.

[5] T. Chen and B. Francis, *Optimal Sampled-data Control Systems*, Springer, New York, 1995.

[6] I. Izadi, T. Chen and Q. Zhao, "Norm invariant discretization for sampled-data fault detection", *Automatica*, vol. 41, pp. 1633–1637, 2005.

[7] P. Zhang, S.X. Ding, G.Z. Wang and D.H. Zhou, "An FDI approach for sampled-data systems", *Proceedings of the American Control Conference*, pp. 2702–2707, 2001.

[8] I. Izadi, S.L. Shah and T. Chen, "A Direct approach to fault detection in non-uniformly sampled systems", *IFAC World Congress*, 2008.

[9] P. Albertos and A. Crespo, "Real-time control of non-uniformly sampled systems", *Control Eng. Practice*, vol. 7, pp. 445–458, 1999.

[10] W. Li, Z. Han and S.L. Shah, "Subspace identification for FDI in systems with non-uniformly sampled multirate data", *Automatica*, vol. 42, pp 619–627, 2006.

[11] W. Li, S.L. Shah and D. Xiao, "Kalman Filters in Non-uniformly Sampled Multirate Systems: for FDI and beyond", *Automatica*, vol. 44, pp 199–208, 2008.

[12] E.Y. Chow and A.S. Willsky, "Analytical redundancy and the design of robust failure detection systems", *IEEE Transactions on Automatic Control*, vol. 29, pp. 603–614, 1984.