Data-Driven Design of Model-based Fault Diagnosis Systems

S. X. Ding

Abstract: In this paper, recent development of data-driven design of fault detection and isolation (FDI) systems is presented. The major attention and focus are on the design schemes for observer-based FDI systems.

Keywords: Fault detection and diagnosis; data-driven methods; observer-based methods; subspace identification methods; parity space methods

1. INTRODUCTION

Process monitoring and fault diagnosis are currently receiving considerably increasing attention in the application and research domains. In the monographs by Gertler [1998], Mangoubi [1998], Chen and Patton [1999], Patton et al. [2000], Russell et al. [2000], Gustafsson [2000], Chiang et al. [2001], Blanke et al. [2003], Simani et al. [2003], Isermann [2000], Ding [2008], both the data-driven and model-based schemes and methods are presented and well described. The recent surveys by Venkatasubramanian et al. [2003a,b], Qin [2003], Zhang and Ding [2008], Mangoubi et al. [2009], Qin [2009], Hwang et al. [2010] provide the reader with the excellent review of the current development of the advanced and sophisticated fault diagnosis schemes and their applications.

As reported in (Venkatasubramanian et al. [2003a], Qin [2003, 2009]), data-driven techniques are widely applied in the process industry for fault detection and isolation (FDI). Among numerous data-driven FDI schemes, the multivariate analysis technique with PCA (principal component analysis) and PLS (partial least squares) as its representative methods is, thanks to its simplicity and efficiency in processing huge amount of process data, recognized as a powerful tool for addressing statistical process monitoring and FDI problems. Although dynamic PCA/PLS (Russell et al. [2000], Li and Qin [2001]), recursive implementation of PCA/PLS (Li et al. [2000], Qin [1998]), fast moving window PCA (Wang et al. [2005]) have been developed in recent years, the multivariate analysis technique seems only efficient in dealing with dynamic processes in the steady state and at higher levels in large-scale systems. In comparison, the model-based FDI technique, thanks to the application of advanced system and control theory, provides a more efficient and powerful tool to investigate FDI issues in highly dynamic systems and control loops which are generally located at the process level. The possible high FDI performance is often achieved at the cost of a highly complex process modelling and, based on it, a sophisticated FDI system design.

Recently, Qin and Li [2001], Ding et al. [2009b], Dong and Verhaegen [2009], Kulcsar et al. [2009], Dong [2009] have proposed SIM (subspace identification methods) based FDI design schemes, in which the FDI systems are directly designed utilizing the collected process data without explicitly identifying a system model. Some of these results have been successfully applied in the industrial processes and also extended for instance to the linear parameter varying systems. The major advantage of these methods is that they can provide high FDI performance similar to the model-based FDI methods but without a sophisticated system design.

The objective of this paper is to present recent development of the data-driven design of FDI systems that are applicable for dynamic processes. Our major attention and focus are on the so-called observer-based FDI systems, which provide not only high FDI performance but also high real-time ability. The paper is organized as follows. A review of the parameterization of the observer-based FDI systems is first presented in Section 2, based on which basic ideas and requirements of data-driven design of FDI systems for dynamic processes are formulated. Section 3 is dedicated to the data-driven schemes for open-loop configured FDI systems, while Section 4 addresses data-driven design of observer-based (and thus closed-loop structured) FDI systems. In Sections 5 and 6, adaptive, fault isolation and identifications issues are briefly addressed.

Notation: The notation adopted throughout this paper is fairly standard. We use \( Q(i : j, p : q) \) denoting the sub-matrix consisting of the \( i \)-th to the \( j \)-th rows and the \( p \)-th to the \( q \)-th columns of \( Q \). \( || \cdot ||_F \) is used for the Frobenius norm of a matrix. \( \mathcal{E}(\cdot) \) represents mean value. \( x \sim \mathcal{N}(0, \Sigma) \) means that \( x \) is normal distributed with zero mean and covariance (matrix) \( \Sigma \). \( X^2(l) \) stands for \( X^2 \)-distribution with \( l \) degrees of freedom and, associated to it, \( \text{prob}(\chi > \chi_{\alpha}(l)) = \alpha \) for the probability of \( \chi > \chi_{\alpha}(l) \) equal to \( \alpha \) (significance level).

2. BASIC IDEAS AND PROBLEM FORMULATION

In this section, we first review observer-based fault detection schemes briefly. It will motivate the problem formulation for the development of data-driven design of model- and observer-based FDI systems.
2.1 Reviewing observer-based fault detection schemes

Consider a process modelled by
\[ x(k+1) = Ax(k) + Bu(k) + w(k) \]  
(1)
\[ y(k) = Cx(k) + Du(k) + v(k) \]  
(2)
where \( u \in \mathbb{R}^l \), \( y \in \mathbb{R}^m \) and \( x \in \mathbb{R}^n \) represent the process input, output and state variable vectors, respectively. \( w \in \mathbb{R}^m \) denote noise sequences that are normally distributed and statistically independent of \( u \) and \( x(0) \).

We assume that the above process model is observable. Let \( G_u(z) = C(zI - A)^{-1}B + D \) and \( v \in \mathbb{R}^m \) be the data model described by
\[ Y_{k,s} = \Gamma_sX_k + Hu,sU_{k,s} + Hw,sW_{k,s} + V_{k,s} \]  
(15)
\[ \Gamma_s = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{s-1} \end{bmatrix} \in \mathbb{R}^{(s+1)m \times n} \]

This motivates us to address the data-driven design of FDI systems in a framework, which can be schematically formulated as solving an (algebraic) equation described by
\[ \forall U, \Psi \begin{bmatrix} Y \\ U \end{bmatrix} = 0 \]  
(10)
with \( Y, U \) denoting the (recorded) process output and input data sets and \( \Psi \) the data-driven realization of the LCF and so that the residual generator to be designed.

This is an alternative way of designing a residual generator directly. A further important issue is the realization of the residual generator \( \Psi \) in a closed-loop configuration and in the (recursive) form of an FDF aiming at ensuring the required robustness on the one hand and real-time ability on the other hand. The last issue addressed in this paper is the development of adaptive schemes, which are used to enhance the robustness of the FDI systems.

3. METHODS OF DATA-DRIVEN DESIGN OF OPEN-LOOP CONFIGURED FD SYSTEMS

In this section, we present three data-driven methods for the design of FDI systems which are configured in an open-loop structure and realized in form of an FIR (finite impulse response) filter. The first two deal with the identification of \( I/O \) models, while the third one is based on a direct identification of the LCF in the form of (10).

3.1 I/O data models

For the purpose of data-driven design of FDI systems, an \( I/O \) model that describes the relation between the input and output data sets plays an essential role. Let \( \omega(k) \in \mathbb{R}^\xi \) be a data vector. We introduce the following notations
\[ \omega_s(k) = \begin{bmatrix} \omega(k) \\ \vdots \\ \omega(k+s) \end{bmatrix} \in \mathbb{R}^{(s+1)\xi} \]  
(11)
\[ \Omega_k = [\omega(k) \cdots \omega(k+N-1)] \in \mathbb{R}^{\xi \times N} \]  
(12)
\[ \Omega_{k,s} = [\omega_s(k) \cdots \omega_s(k+N-1)] \]  
(13)
where \( s, N \) are some (large) integer (Qin [2006], Huang and Kadali [2008]). In our study, \( \omega(k) \) can be \( y(k), u(k), z(k) \), and \( \xi \) represents \( m \) or \( l \) or \( n \) given in (1)-(2).

The first I/O data model described by
\[ Y_{k,s} = \Gamma_sX_k + H_{u,s}U_{k,s} + H_{w,s}W_{k,s} + V_{k,s} \]  
(15)
\[ \Gamma_s = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{s-1} \end{bmatrix} \in \mathbb{R}^{(s+1)m \times n} \]
follows directly from (1)-(2), where $H_{u,s} W_{k,s} + V_{k,s}$ represents the influence of the noise vectors on the process output with $H_{u,s}$ having the same structure like $H_{u,s}$ and $W_{k,s}, V_{k,s}$ as defined in (11)-(13).

In the SIM framework, the so-called innovation form, instead of (1)-(2), is often applied to build an I/O model (Qin [2006], Huang and Kadali [2008]). The core of the innovation form is a Kalman filter, which is written as

$$
\hat{x}(k+1) = A\hat{x}(k) + Bu(k) + Ke(k) \quad (16)
$$

with the innovation $y(k) - \hat{y}(k) := e(k)$ being a white noise sequence and $K$ the Kalman filter gain matrix. Based on (16)-(17), the I/O relation of the plant can be alternatively written into

$$
\hat{x}(k+1) = A\hat{x}(k) + Bu(k) + Ke(k) \iff \hat{x}(k+1) = A\hat{x}(k) + B\hat{u}(k) + K\hat{y}(k) \quad (19)
$$

and

$$
y(k) = C\hat{x}(k) + Du(k) + e(k) \quad (20)
$$

The following two I/O data models follow respectively from (18), (20) and (19), (20):

$$
Y_{k,s} = \Gamma_{s} \hat{x}_{k} + H_{u,s} u_{k,s} + H_{e,s} e_{k,s} \tag{21}
$$

$$
H_{e,s} = \begin{bmatrix}
D & 0 & \cdots & 0 \\
CB & 0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
CA^{s-1} B & \cdots & CB D
\end{bmatrix} \in \mathcal{R}^{s+1\times (s+1)}
$$

Since the eigenvalues of $A_{K}$ are all located within the unit circle, for a large integer $\rho$

$$
A_{K}^{\rho} \approx 0 \implies \hat{x}(k) \approx \sum_{i=1}^{\rho} A_{K}^{-1} [ B \ K ] \begin{bmatrix} u(k-i) \\ y(k-i) \end{bmatrix} \tag{23}
$$

As a result, (21) can be re-written into

$$
Y_{k,s} \approx \Upsilon_{s,\rho \to 1} Z_{k-\rho \to 1} + H_{u,s} u_{k,s} + H_{e,s} e_{k,s} \tag{24}
$$

$$
Z_{k-\rho \to 1} = \begin{bmatrix} z_{-\rho \to 1} \cdots \cdots z_{1-\rho \to 1} (k-\rho + N) \end{bmatrix} \begin{bmatrix} u(k-\rho) \\ y(k-\rho) \\ \vdots \\ u(k-1) \\ y(k-1) \end{bmatrix}
$$

$$
\Upsilon_{s,\rho \to 1} = \begin{bmatrix} A_{K}^{-1} [ B \ K ] & \cdots & [ B \ K ] \end{bmatrix}
$$

Now, identifying $\Upsilon_{s,\rho \to 1}, H_{u,s}$ using process data $Y_{k,s}, U_{k,s}, Z_{k-\rho \to 1}$ (Huang and Kadali [2008]) allows the construction of a residual generator and further the computation of the covariance matrix of the residual vector as well as the threshold setting, as summarized in the following algorithm.

\textbf{Algorithm 3.1: FIR FD system design}

\textbf{S1:} Collect process data and build $Z_{k-\rho \to 1}, U_{k,s}, Y_{k,s}$

\textbf{S2:} Solve the least squares problem for $\Upsilon_{s,\rho \to 1}, H_{u,s}$

\textbf{S3:} Compute $\Sigma_{r} = \frac{1}{T} \sum_{t=0}^{T} e_{s,t} e_{s,t}^{T}$

\textbf{S4:} Set the threshold $J_{th} = \sqrt{\Sigma_{r}} ((s+1)m)$

In the above algorithm,

$$
H_{e,s} e_{k,s} = Y_{k,s} - \Upsilon_{s,\rho \to 1} Z_{k-\rho \to 1} - H_{u,s} u_{k,s}
$$

with $\Upsilon_{s,\rho \to 1}$ and $H_{u,s}$ being identified in S2, which are then applied to the residual generation $r_{s}(k+\rho) = y_{s}(k+\rho) - \Upsilon_{s,\rho \to 1} Z_{k-\rho \to 1}(k) - H_{u,s} u_{s}(k+\rho) \tag{25}$

\textbf{Scheme II} This scheme has been proposed by Dong [2009]. It is based on the I/O model (22), which is further re-written into, using (23),

$$
Y_{k,s} = \Upsilon_{s,\rho \to 1} Z_{k-\rho \to 1} + H_{u,s} u_{k,s} + H_{e,s} e_{k,s} \tag{26}
$$

$$
T_{s,\rho \to 1} = \Gamma_{s}^{-1} \begin{bmatrix} A_{K}^{-1} [ B \ K ] & \cdots & [ B \ K ] \end{bmatrix}
$$

Having constructed $\Upsilon_{s,\rho \to 1}, H_{u,s}, H_{e,s}$ using the Markov parameters

$$
[ C A_{K}^{-1} B \ C A_{K}^{-1} K \cdots CB \ CD ] := \Xi, \kappa = s + \rho
$$

which is identified applying the model

$$
Y_{k} \approx \sum_{i=0}^{\infty} [ C A_{K}^{-1} [ B \ K ] \begin{bmatrix} U_{k-1-i} \\ Y_{k-1-i} \end{bmatrix} + DU_{k} + E_{k}
$$

$$
Y_{k-i} = \begin{bmatrix} y(k-i) \cdots y(k+i+N) \end{bmatrix} \in \mathcal{R}^{m \times (N+1)}
$$

$$
U_{k-i} = \begin{bmatrix} u(k-i) \cdots u(k+i+N) \end{bmatrix} \in \mathcal{R}^{m \times (N+1)}
$$

$$
i = 0, \cdots, s, E_{k} = [ e(k) \cdots e(k+N) ]
$$

residual vector can then be generated by

$$
r_{s}(k+\rho) = (I - H_{K}^{s}) y_{s}(k+\rho) - \Upsilon_{\rho \to 1} Z_{k,h+\rho-1} + H_{u,s} u_{s}(k+\rho) \tag{27}
$$
Algorithm 3.2: FD system design (Dong [2009])

S1: Collect process data and build $Z_{k-s,p,s-1}, U_{k,s}, Y_{k,s}$

S2: Solve the least squares problem for $Z$

$$\min_{Z} \left\| Y_k - \begin{bmatrix} Z_{k-s,p,s-1} \\ U_{k,s} \end{bmatrix} \right\|_F$$

S3: Form $H_{y,s} \subset \mathbb{K}^{p,1}$, $H_{u,s} \subset \mathbb{K}$ using $Z$

S4: Compute $\Sigma = \begin{bmatrix} \frac{1}{n} E_k^s \end{bmatrix} E_{y,s}^T$

$$E_{y,s} = (I - H_{y,s}^K) Y_{k,s} - Y_{k,s}^{\perp} H_{y,s}^{-1} + H_{u,s} U_{k,s}$$

S5: Set the threshold $J_{th} = \chi^2_{0.05}(s+1)$

It is worth to remark that in the above two algorithms the least squares estimations using process data sets build the major computations. As pointed out in (Huang and Kadali [2008]), they can be implemented in a numerically robust way with a QR-decomposition of the process data as follows

$$\begin{bmatrix} Z_{k-s,p,s-1} \\ U_{k,s} \\ Y_{k,s} \end{bmatrix} = \begin{bmatrix} R_{11} & 0 & 0 \\ R_{21} & R_{22} & 0 \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \end{bmatrix}$$

where $s_p$ is some large integer. For instance, $[Y_{s,p-1}, H_{u,s}]$ in Algorithm 3.1 can be determined by

$$[Y_{s,p-1}, H_{u,s}] = \begin{bmatrix} R_{31} \end{bmatrix} \begin{bmatrix} R_{32} \end{bmatrix}^{-1}$$

Moreover, the relation (Qin [2006])

$$R_{33} Q_3 = H_{y,s} E_{y,s} = H_{y,s} W_{k,s} + V_{k,s}$$

can be used for the determination of the covariance matrix needed for the threshold setting.

**Scheme III**

Inspired by the work in Qin and Li [2001], Wang and Qin [2002], Ding et al. [2009b], the idea behind this scheme is the identification of the data-driven form of the LCF as described in (10). To this end, re-write (15) or (21) into

$$\begin{bmatrix} U_{k,s} \\ Y_{k,s} \end{bmatrix} = \Psi_s \begin{bmatrix} U_{k,s} \\ Y_{k,s} \end{bmatrix} + H_{u,s} W_{k,s} + V_{k,s}$$

where $s_p$ is some large integer. For instance, $[Y_{s,p-1}, H_{u,s}]$ in Algorithm 3.1 can be determined by

$$[Y_{s,p-1}, H_{u,s}] = \begin{bmatrix} R_{31} \end{bmatrix} \begin{bmatrix} R_{32} \end{bmatrix}^{-1}$$

Moreover, the relation (Qin [2006])

$$R_{33} Q_3 = H_{y,s} E_{y,s} = H_{y,s} W_{k,s} + V_{k,s}$$

can be used for the determination of the covariance matrix needed for the threshold setting.

It is worth to remark that in the above two algorithms the least squares estimations using process data sets build the major computations. As pointed out in (Huang and Kadali [2008]), they can be implemented in a numerically robust way with a QR-decomposition of the process data as follows

$$\begin{bmatrix} Z_{k-s,p,s-1} \\ U_{k,s} \\ Y_{k,s} \end{bmatrix} = \begin{bmatrix} R_{11} & 0 & 0 \\ R_{21} & R_{22} & 0 \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \end{bmatrix}$$

where $s_p$ is some large integer. For instance, $[Y_{s,p-1}, H_{u,s}]$ in Algorithm 3.1 can be determined by

$$[Y_{s,p-1}, H_{u,s}] = \begin{bmatrix} R_{31} \end{bmatrix} \begin{bmatrix} R_{32} \end{bmatrix}^{-1}$$

Moreover, the relation (Qin [2006])

$$R_{33} Q_3 = H_{y,s} E_{y,s} = H_{y,s} W_{k,s} + V_{k,s}$$

can be used for the determination of the covariance matrix needed for the threshold setting.

It is worth to remark that in the above two algorithms the least squares estimations using process data sets build the major computations. As pointed out in (Huang and Kadali [2008]), they can be implemented in a numerically robust way with a QR-decomposition of the process data as follows

$$\begin{bmatrix} Z_{k-s,p,s-1} \\ U_{k,s} \\ Y_{k,s} \end{bmatrix} = \begin{bmatrix} R_{11} & 0 & 0 \\ R_{21} & R_{22} & 0 \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \end{bmatrix}$$

where $s_p$ is some large integer. For instance, $[Y_{s,p-1}, H_{u,s}]$ in Algorithm 3.1 can be determined by

$$[Y_{s,p-1}, H_{u,s}] = \begin{bmatrix} R_{31} \end{bmatrix} \begin{bmatrix} R_{32} \end{bmatrix}^{-1}$$

Moreover, the relation (Qin [2006])

$$R_{33} Q_3 = H_{y,s} E_{y,s} = H_{y,s} W_{k,s} + V_{k,s}$$

can be used for the determination of the covariance matrix needed for the threshold setting.
solve Luenberger equations
\[ TA - A_z T = L_z C \]  
(39)
\[ c_z T = g C, c_z = \begin{bmatrix} 0 & \cdots & 0 & 1 \end{bmatrix}, g = \alpha_{s,s} \]  
(40)

A direct application of this result is the construction of an observer-based residual generator for the given parity vector \( \alpha \) as follows
\[ z(k+1) = A_z z(k) + B_z u(k) + L_z y(k) \in \mathbb{R}^s \]  
(41)
\[ r(k) = g y(k) - c_z z(k) - d_z u(k) \in \mathbb{R} \]  
(42)
\[ B_z = TB - L_z D, d_z = g D \]  
(43)

Note that
\[
\begin{bmatrix} B_z \\ d_z \end{bmatrix} = \begin{bmatrix} \alpha_{s,0} & \alpha_{s,2} & \cdots & \alpha_{s,s-1} & \alpha_{s,s} \\ \alpha_{s,1} & \cdots & \cdots & \cdots & 0 \\ \vdots & \cdots & \cdots & \cdots & \vdots \\ \alpha_{s,s} & 0 & \cdots & \cdots & 0 \end{bmatrix} \begin{bmatrix} D \\ CB \\ CAB \\ CA^{-1}B \end{bmatrix}
\]  
(44)

Equations (37), (40) and (44) allow a direct construction of an observer-based residual generator given by (41)-(42) using a row of \( \Psi_+ \) and without any additional computation and design efforts.

Remember that \( s \) is generally selected sufficiently large (typically much larger than \( n \)) during the identification phase. From the real-time computational viewpoint, it is of practical advantage to reduce the order of the residual generator as much as possible. For this purpose, Ding et al. [2009b] have proposed the following algorithm.

**Algorithm 4.1: Order reduction**

**S1:** Do a QR decomposition of \( \Gamma_s^{-1} V = [Q_1 R_1 \quad Q_2] \)

with \( V = \begin{bmatrix} 0 & \cdots & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \in \mathbb{R}^{m(s+1) \times m(s+1)} \)

\( R_1 \) being a \( \eta \times \eta \) upper triangular, \( Q_1 \) a \( \eta \times \eta \) with orthonormal columns and \( Q_2 \) a \( \eta \times n \) matrix

**S2:** Compute \( Q_1^{-1} \Gamma_s^{-1} = [R_1 \quad Q_1^{-1} Q_2] V^{-1} \)

As proven in (Ding et al. [2009b], Theorem 1), the result of the above algorithm is
\[ Q_1^{-1} \Gamma_s^{-1} = [Q_3 \quad \Psi] \]  
(45)

where \( \eta = m(s+1) - n, Q_3 \) is a \( \eta \times \eta \) matrix and \( \Psi \) is of the form
\[
\Psi = \begin{bmatrix} \psi_{1,1} & \cdots & \psi_{1,\eta-2} & \psi_{1,\eta-1} & \psi_{1,\eta} \\ \psi_{2,1} & \cdots & \psi_{2,\eta-2} & \psi_{2,\eta-1} & 0 \\ \vdots & \cdots & \cdots & \cdots & \vdots \\ \psi_{\eta-1,1} & \psi_{\eta-1,2} & 0 & \cdots & 0 \\ \psi_{\eta,1} & 0 & 0 & \cdots & 0 \end{bmatrix} \in \mathbb{R}^{\eta \times \eta} 
\]  
(46)

It is evident that the last row of \( Q_1^{-1} \Gamma_s^{-1} \) is of the form
\[ [\alpha_0 0 \cdots 0] \in \mathbb{R}^\eta \]

and thus \( \alpha_n \in \mathbb{R}^{(n+1)m} \) is a parity vector whose order is \( n \).

Recall that \( \Psi_\perp \) delivered by Algorithm 3.3 is composed of \( \Psi_{s,y} \perp \psi_{s,u} \) with \( \psi_{s,y} ^\perp = \Gamma_s ^\perp \). Applying Algorithm OR to \( \psi_{s,u} \) yields \( Q_1^{-1} \psi_{s,u} \). Now, multiply \( Q_1^{-1} \) to \( \psi_{s,u} \) and denote the last row of \( Q_1^{-1} \psi_{s,u} \) by \( \alpha_{n,u} \). It holds, following (44),
\[ \alpha_{n,u} = \alpha_n H_{s,u} \Rightarrow \begin{bmatrix} B_z \\ d_z \end{bmatrix} = \begin{bmatrix} \alpha_{n,u}(1 : 1) \\ \alpha_{n,u}(l + 1 : 2l) \\ \vdots \\ \alpha_{n,u}(nl + 1 : (n + 1)l) \end{bmatrix} 
\]  
(47)

In summary, we have

**Algorithm 4.2: Observer-based FD system design**

**S1:** Run Algorithm 3.3 with output \( \Psi_{s,y} \perp \psi_{s,u} \)

**S2:** Run Algorithm 4.1

**S3:** Compute \( Q_1^{-1} \psi_{s,u} \) for \( \alpha_{n,u} \)

**S4:** Form \( B_z, d_z, L_z \) according to (47), (38)

**S5:** Construct residual generator (41)-(42)

Although Algorithm 4.2 only delivers a scalar residual signal, it is fundamental for the development of multiple residual generation. It is well-known that an m-dimensional residual vector is necessary for a reliable fault detection and isolation in the framework of observer-based FDI systems (Ding [2008]). For this purpose, \( m \) vectors should be selected from \( \Psi_+ \). Due to the space limitation, this topic cannot be addressed in this paper. We refer the reader to (Ding et al. [2011, 2012]) for some recent results on this topic.

### 4.2 Data-driven design of Kalman filter based FD systems

It is well known that a Kalman filter (18)-(20) delivers an innovation \( e(k) = y(k) - \hat{y}(k) \) with
\[
\mathbb{E} (e(i)e^T(j)) = \Sigma_i \delta_{ij} 
\]  
(48)

and builds, together with test statistic \( J = e^T(k) \Sigma_e ^{-1} e(k) \) and the threshold \( J_T = \chi^2_m \) an FD system with the optimal trade-off between the false alarm rate and fault detectability. This sub-section addresses (i) identification of the Kalman filter gain matrix \( K \) and the covariance matrix \( \Sigma_e \) based on the I/O model (21) and Algorithm 3.3 (ii) construction of a Kalman filter-based FD system.

Given \( \Psi_s ^\perp, \psi_{s,y} ^\perp Z_{k,s} = \psi_{s,y} ^\perp R_{33} Q_3 = \psi_{s,y} ^\perp H_{e,s} E_{k,s} \) delivered by Algorithm 3.3. It turns out
\[
\frac{1}{N} E_{k,s} Y_k ^T \approx \begin{bmatrix} \Sigma_e \\ 0 \\ 0 \end{bmatrix} \implies \frac{1}{N} \Psi_{s,y} ^\perp Z_{k,s} Y_k ^T = 
\]  
(49)

\[
\frac{1}{N} \Psi_{s,y} ^\perp H_{e,s} E_{k,s} Y_k ^T \approx \psi_{s,y} ^\perp \begin{bmatrix} \Sigma_e \\ C \Sigma_e \\ C \Lambda \Sigma_e \end{bmatrix} 
\]  
(49)

Now, solve
\[ \Psi_{s,y} ^\perp \Gamma_s = 0 \]  
(50)

for \( \Gamma_s \) and further compute...
Also an SVD. Motivated by this observation, Naik et al. [2010] proposed two schemes for recursively updating the identification of parity vectors. One of them is based on the first-order perturbation (FOP) theory, which is a rank-one update of the eigenpairs of the data covariance matrix, and another one on the data-projection method (DPM) which serves as a simple approach for recursively updating the singular values and associated eigenvectors of interests. These two schemes can be used either to construct an (open-loop) adaptive residual generator of the form (36) or for the implementation of an adaptive observer-based residual generator. In this section, we present an approach proposed in (Ding et al. [2009a]) to the design and implementation of an adaptive residual generator. Different from the above-mentioned methods, the basic idea of this approach is the application of the well-established adaptive control methods (Åström and Wittenmark [1995]) to the observer-based residual generator.

5.1 Problem formulation

For the simplicity of our discussion, we only consider the residual generator (41)-(42) designed using Algorithm 4.2. It is evident that parameter changes in the original system matrices are now represented by the changes in $B_z, L_z, g, d_z$. In the sequel, we assume that there exists no structural change, i.e. change e.g. in the observability, and the changes in parameters are slow and can be considered nearly constant in a (large) time interval. For our purpose, we now extend the residual generator (41)-(42) to

\[
\begin{align*}
\dot{z}(k+1) &= A_z z(k) + B_z u(k) + T K e(k) \\
r(k) &= c(k) = y(k) - C_z z(k) - D_z u(k) \\
A_z &= A_z + L_z C_z, B_z = B_z + L_z D_z
\end{align*}
\]

It is straightforward to prove that the dynamics of the residual generator is governed by

\[
\begin{align*}
\Delta x(k+1) &= A_K \Delta x(k) + T w(k) - T K v(k) \\
\Delta x(k) &= T x_z(k) - z(k), A_K = T (A - K C) T^{-1}
\end{align*}
\]

Considering that $c(k) \sim N(0, \Sigma_c)$, the following statistic and threshold are applied for the fault detection purpose

\[
J = r^T(k) \Sigma_c^{-1} r(k), J_{th} = \chi^2_{\alpha}(m)
\]

Algorithm 4.3. Kalman filter based FD

1. Run Algorithm 3.3
2. Compute $\Psi_{s,y} R_{ss} Q_s Y_k^T$
3. Solve (50) and (51)
4. Generate an $m$-dimensional residual vector
5. Compute (52)

5. THE ADAPTIVE SCHEME

Among the existing adaptive schemes, the recursive technique has been well developed and widely applied to the standard data-driven methods like PCA, PLS. Since the early work by Helland et al. [1992], Qin [1998], numerous recursive schemes have been reported. The recent research focus is mainly on the recursive computation of SVD which is used both in the PCA and PLS. Li et al. [2000] proposed to apply the rank-one modification technique for a recursive updating of the covariance matrix and its SVD used in the PCA. It is remarkable that the key step in Algorithm 3.3 and all those associated algorithms is also an SVD. Motivated by this observation, Naik et al. [2010] proposed two schemes for recursively updating the identification of parity vectors. One of them is based on the first-order perturbation (FOP) theory, which is a rank-one update of the eigenpairs of the data covariance matrix, and another one on the data-projection method (DPM) which serves as a simple approach for recursively updating the singular values and associated eigenvectors of interests. These two schemes can be used either to construct an (open-loop) adaptive residual generator of the form (36) or for the implementation of an adaptive observer-based residual generator. In this section, we present an approach proposed in (Ding et al. [2009a]) to the design and implementation of an adaptive residual generator. Different from the above-mentioned methods, the basic idea of this approach is the application of the well-established adaptive control methods (Åström and Wittenmark [1995]) to the observer-based residual generator.

5.1 Problem formulation

For the simplicity of our discussion, we only consider the residual generator (41)-(42) designed using Algorithm 4.2. It is evident that parameter changes in the original system matrices are now represented by the changes in $B_z, L_z, g, d_z$. In the sequel, we assume that there exists no structural change, i.e. change e.g. in the observability, and the changes in parameters are slow and can be considered nearly constant in a (large) time interval. For our purpose, we now extend the residual generator (41)-(42) to

\[
\begin{align*}
\dot{z}(k+1) &= A_z z(k) + B_z u(k) + T L z y(k) + L o r(k) \\
r(k) &= g y(k) - c_z z(k) - d_z u(k)
\end{align*}
\]

where $L_o$ provides additional degree of design freedom and should ensure the stability of $A_z = A_z - L_o e_z$. Let

\[
\theta = \begin{bmatrix} \theta_u \\ \theta_y \end{bmatrix} \in \mathbb{R}^{(s+1)(m+1)}, \theta_u = \text{col} \left[ B_z \right], \theta_y = \text{col} \left[ \begin{bmatrix} L \\ g \end{bmatrix} \right]
\]

$Q(u(k), y(k)) = \left[ U(k) - L_o u^T(k) \right] \left[ y(k) - L o y^T(k) \right]$

\[
U(k) = \left[ u_1(k) \times I_{s \times s} \cdots u_{m}(k) \times I_{s \times s} \right]
\]

\[
Q(k) = \left[ y_1(k) \times I_{s \times s} \cdots y_m(k) \times I_{s \times s} \right]
\]

(57)-(58) can be re-written into

\[
\begin{align*}
\dot{z}(k+1) &= A_z z(k) + Q(u(k), y(k)) \theta \\
r(k) &= \left[ -u^T(k) y^T(k) \right] \begin{bmatrix} d_k^T \\ g_k^T \end{bmatrix} - c_z z(k)
\end{align*}
\]

The task consists in designing a residual generator which is adaptive to $\theta$ and delivers a residual signal $r(k)$ satisfying $\lim_{k \to \infty} r(k) = 0$ and, if possible, with an exponential converging speed independent of a constant change in $\theta$.

5.2 The adaptive residual generator algorithm

The adaptive residual generator scheme given in (Ding et al. [2009a]) is inspired by the adaptive observer schemes proposed by Zhang [2002]. It consists of three sub-systems:

Residual generator

\[
\dot{z}(k+1) = A_z z(k) + Q(u(k), y(k)) \hat{\theta}(k) + V(k+1) \left( \hat{\theta}(k+1) - \hat{\theta}(k) \right) \\
r(k) = \left[ -u^T(k) y^T(k) \right] \begin{bmatrix} d_k^T \\ g_k^T \end{bmatrix} - c_z z(k)
\]
Auxiliary filter
\[ V(k+1) = \bar{A}_z V(k) + Q(u(k), y(k)) \quad (63) \]
\[ \varphi(k) = c_z V(k) - \left[ \begin{array}{c} 0 \\ \vdots \\ -u^T(k) \end{array} \right] \left[ \begin{array}{c} 0 \\ \vdots \\ y^T(k) \end{array} \right] \quad (64) \]

Parameter estimator
\[ \hat{\theta}(k+1) = \gamma(k) \varphi^T(k) r(k) + \hat{\theta}(k) \quad (65) \]
\[ \gamma(k) = \frac{\mu}{\delta + \varphi(k) \varphi^T(k)}, \delta \geq 0, 0 < \mu < 2 \quad (66) \]

Algorithm 5.1: Adaptive residual generator

\[ S_0: \text{Set the initial values } k = 0, z(0), \theta(0), V(0) = 0 \]
\[ \varphi(0) = 0, r(0) = \tilde{g}(0) y(0) - c_z \hat{z}(0) - \tilde{d}(0) u(0) \]

\[ S_1: \text{Compute } V(k+1) \text{ according to (63)} \]
\[ S_2: \text{Compute } \hat{\theta}(k+1) \text{ according to (65)} \]
\[ S_3: \text{Compute } \hat{z}(k+1) \text{ according to (61)} \]
\[ S_4: \text{Increase } k \text{ by one, receive } y(k), u(k) \]
\[ S_5: \text{Compute } r(k), \varphi(k) \text{ according to (62) and (64)} \]

5.3 Stability and exponential convergence

The major advantage of applying the above adaptive technique is that the convergence of the parameter estimation and the whole system stability are guaranteed. In order to demonstrate it, let \( \eta(k) = \hat{z}(k) - V(k) \hat{\theta}(k) \) with
\[ \hat{z}(k) = z(k) - \hat{z}(k), \hat{\theta}(k) = \theta - \hat{\theta}(k) \]

Then \( r(k) \) can be re-written into
\[ r(k) = \left[ -u^T(k) y^T(k) \right] \left[ \begin{array}{c} d^T_{2c}(k) \\ \hat{g}^T_{2c}(k) \end{array} \right] - c_z \hat{z}(k) \]
\[ = g y(k) - c_z \hat{z}(k) - \tilde{d}_z(k) u(k) - (g - \tilde{g}(k)) y(k) \]
\[ = c_z \hat{z}(k) - \left[ -u^T(k) y^T(k) \right] \left[ \begin{array}{c} d^T_{2c}(k) \\ \hat{g}^T_{2c}(k) \end{array} \right] \]
\[ \hat{d}_z(k) = d_z - \tilde{d}_z(k), \tilde{g}(k) = g - \tilde{g}(k) \]

Moreover,
\[ \hat{\theta}(k+1) = \Theta(k) \eta(k) + (I - \gamma(k) \varphi^T(k) \varphi(k)) \hat{\theta}(k) \quad (68) \]
\[ \Theta(k) = -\gamma(k) \varphi^T(k) c_z \]
\[ \eta(k+1) = A_z \eta(k) \quad (70) \]

Based on (67)-(70) and the results given in (Aström and Wittenmark [1995], Zhang [2002]), Ding et al. [2009a] have proven that \( \lim_{k \to \infty} r(k) = 0 \).

6. ON FAULT ISOLATION AND IDENTIFICATION

In practice of (statistical) process monitoring, contribution plots are widely applied as a standard PCA-based fault isolation and identification technique (Chiang et al. [2001]). Recently, Qin [2009], Alcala and Qin [2009], Li et al. [2010] have proposed to estimate/identify the faults in a statistical process by means of reconstruction. In (Dong [2009]), a data-driven approach has been developed to identify faults in a dynamic process. With the aid of the parity space identified using Algorithm 3.3, the algebraic fault isolation methods e.g. given in (Ding [2008]) can also be used. It is remarkable that these approaches have been developed on the assumption of an available fault model. Alternatively, Ding et al. [2009b], Wang et al. [2011] have proposed data-driven design schemes for the isolation of sensor and actuator faults in dynamic systems.

In practice, data collected during faulty process operations are seldom available or at least often not available in a well synchronized form with a detailed fault description. It makes a data-driven (fault) modelling or data-driven fault isolation and identification very difficult. Data-driven fault isolation and identification is a changing topic both in the application and research domains. Due to the space limitation, it will not be addressed in this paper.

7. CONCLUSIONS

In this paper, we have presented a number of data-driven methods for the design of FDI systems. The major focus has been on the data-driven design of observer-based FDI systems. Most of the data-driven design algorithms presented in this paper have been realized in the software form and studied on different benchmark processes like Tennessee Eastman (TE) process (Russell et al. [2000]) and continuous stirred tank heater (CSTH) (Thornhill et al. [2008]). Some of them have also been tested on the real laboratory systems including Three-Tank-System and CSTH that are available at AKS. In addition, they have been applied to the identification and monitoring of process key variables and to the data-driven design of feedback control systems. Due to the space limitation, the achieved results cannot be included in this paper.

It is remarkable that the data-driven design technique for dynamic FDI systems is still in its early stage. For a successful application in practice as an established technique, great efforts should be made to address, for instance, the fault isolation and identification issues as mentioned previously and the nonlinear and non-Gaussian issues. The latter is very important and useful for improving the FDI performance during the process operation. The successful methods based on machine learning technique show promising results (Yu and Qin [2008, 2009]).

Acknowledgement: The author would like to thank S. Yin and Y. Wang for their valuable technical support, and the reviewers for their valuable comments.

REFERENCES


S. X. Ding, S. Yin, Y. Wang, Y. Wang, Y. Yang, and B. Ni. Data-driven design of observers and its applications. Preprint of the 18th IFAC World Congress, pages 11441–11446, 2011.


