Control and Cybernetics

vol. **44** (2015) No. 2

A modified independent component analysis-based fault detection method in plant-wide systems*

by

Mazdak Teimoortashloo[1] and Ali Khaki Sedigh[2]

[1]Department of Electrical Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran
mazdak1978@bojnourdiau.ac.ir
[2]Industrial Control Center of Excellence, Department of Electrical and Computer Engineering, K.N. Toosi University of Technology, Tehran, Iran
sedigh@kntu.ac.ir

Abstract: This paper presents a modified Independent Component Analysis (ICA)-based Fault Detection Method (FDM). The proposed FDM constructs a set of matrices, revealing the trend of the variable samples and execute ICA algorithm for each set of matrices in contrast to the FDM based on dynamic ICA (DICA) which constructs the high dimensional augmented matrix. This paper shows that the proposed FDM decreases the matrix dimensions and as a result compensates for some disadvantages of using the high dimensional matrix discussed in previous articles. Furthermore, other advantages of the proposed FDM are the decreases in the running time, computational cost of the algorithm and the orthogonalization estimation errors. Moreover, the proposed method improves the detectability for a class of faults compared to DICA-based FDM. This class of fault occurs when two or more consecutive samples of fault source signal have opposite signs and cancel out each other. Simulation results are provided to show the effectiveness of the proposed methodology.

Keywords: DICA, fault detection, running time, orthogonalization

1. Introduction

In plant-wide systems, there are many variables sampled by sensors all over the plant, leading to large volumes of data. Processing such data requires powerful computational facilities, which may not be available at many industrial sites. Thus, it is necessary to find methods that can extract as much as information as

*Submitted: October 2014; Accepted: June 2015

possible from these data in a short time with low computational burden. Furthermore, in the past decades, the cost of raw material and energy has increased and, due to the popular demand for higher efficiency, the waste exchange lines are installed to preclude raw material loss, and buffer tanks are bypassed to save energy. As a result, these changes engender highly interactive loops in the system. In such systems, the influence of a fault, having occurred in one loop, propagates through all the loops and hence the operator cannot diagnose the root cause of the fault. In these systems, Partial Least Squares (PLS), Principal Component Analysis (PCA) and Independent Component Analysis (ICA) are widely used for monitoring and fault detection and diagnosis (Thornhill and Horch, 2007).

The PLS is a statistical method that finds a linear regression model by projecting variables into a new space (Vinzi, Chin, Henseler and Wang, 2010). A multi-way kernel total projection partial least squares (MKT-PLS) algorithm was proposed to improve FDM accuracy of batch process in Zhao, Xue and Wang (2014). The hybrid kernel T-PLS algorithm was proposed to combine the global and local functions to solve non-linear problems in FDM by Zhao and Xue (2014). A modified kernel PLS (KPLS) was proposed to improve FDM in nonlinear systems by Wang and Shi (2014).

The PCA algorithm maps the space of the observation variables onto a space of new variables, such that the correlations between the principal components (PCs) are minimized, and then the dimension reduction is performed, omitting the PCs corresponding to smaller variances (Isermann, 2006; Lee, Yoo and Lee, 2004a, or Hyvarinen, Karhunen and Oja, 2001). A two step PCA algorithm is suggested for outlier detection in Tharrault, Mourot, Ragot and Maquin (2008). Kernel PCA (KPCA) is proposed as a nonlinear method in Nguyen and Golinval (2010). The statistics kernel PCA (SKPCA) is proposed, meant to reveal more useful information in FDM by using higher order statistics, in Hehe, Yi and Hongbol (2011). In the PCA method, it is assumed that PCs or latent variables are Gaussian. However, it has been shown that latent variables of most of the real systems do not follow a Gaussian distribution (Martin and Morris, 1996).

The ICA-based process monitoring is proposed in Lee, Yoo and Lee (2004a), or in Chun-Chin, Mu-Chen and Long-Sheng (2010). In the ICA method, statistical dependence between simultaneous samples of variables is minimized (Lee, 1998). It is also shown that this method can reveal more useful information from the observations in comparison with the PCA when the latent variables are not Gaussian (Lee, Yoo and Lee, 2004a). However, the system is assumed to be static in the ICA-based FDMs. The approach is extended in Lee, Yoo and Lee (2004b) and a Dynamic ICA (DICA) is proposed to overcome this disadvantage by stacking l lag measurements and reconstructing the augmented observation matrix. The choice of the number of lag measurements (l) is a challenge, too. In Lee, Yoo and Lee (2004a) $l = 2$ is proposed, while an automatic algorithm to determine the number of lags is proposed in Ku, Storer and Georgakis (1995). A particle swarm optimization-ICA is proposed to determine the order of the ICs by Zhang and Zhang (2010). PCA and ICA are combined in order to deal with

Gaussian and non-Gaussian latent variable in Ge and Song (2007). A modified ICA is proposed to extract the dominant ICs in normal operating condition in Lee, Qin and Lee (2006). A method of residual analysis with ICA based on average multivariate cumulative sum (AMRA-ICA) is proposed to avoid the influence of periodic disturbance in non-Gaussian chemical processes in Tian, Du and Qian (2013).

In general, the observation matrix, obtained by FDM, has high dimensionality in the case of monitoring of the plant-wide system. This phenomenon is critical when constructing the augmented matrix. Therefore, to overcome the drawbacks of high dimensional observation matrix, algorithms that construct observation matrix with low dimensions are proposed. The matrix of low dimension improves the robust performance (Lee, Yoo and Lee, 2004a), reduces the complexity of analysis (Lee, Yoo and Lee, 2004a, or Puntonet and Prieto, 2004), improves the sensitivity to whitening (Le, Karpenko, Ngiam and Ng, 2011), and it can be a strategy for dealing with the small sample size problem (Deng, Liu, Hu and Guo, 2012).

The main contribution of this paper is that in comparison to the DICA-based FDM, the here proposed FDM allows for constructing the low dimensional set of matrices. Therefore, it improves detection capabilities by overcoming the drawbacks of dealing with high dimensional systems, which have been discussed above. The other advantages of the proposed FDM, such as reduction in the running time, in the number of arithmetic operations, and in the error of orthogonalization estimation, as well as the overall improvement of capability of fault detection are explained in subsection 3.2.

The rest of the paper is organized as follows. The second section presents the ICA algorithm and the DICA-based FDM. Then, the third section introduces the proposed FDM algorithm and its advantages. The fourth section validates the advantages of the proposed FDM by simulation. Conclusion and appendix form the last sections of this paper.

2. The ICA algorithm and the DICA-based fault detection

The idea of the ICA algorithm is to find m ICs as sources ($\mathbf{s}(k) \in R^{m\times 1}$) from the m observed variables ($\mathbf{x}(k) \in R^{m\times 1}$) that are sampled from the sensors. They construct the k^{th} vectors of the matrices $\mathbf{S} \in R^{m\times N}$ and $\mathbf{X} \in R^{m\times N}$, respectively. Actually, this algorithm attempts to find the mixing matrix $\mathbf{A} \in R^{m\times m}$ and the IC matrix $\mathbf{S} \in R^{m\times N}$, which are unknown matrices, from the known observation matrix $\mathbf{X} \in R^{m\times N}$, where

$$\mathbf{X} = \mathbf{AS} \tag{1}$$

and N is the number of samples of each variable. Equation (1) can be rewritten as $\mathbf{S} = \mathbf{WX}$ where $\mathbf{W} = \mathbf{A}^{-1}$ is called the de-mixing matrix. In order to execute the ICA algorithm, matrix must be whitened using $\mathbf{Q}$, where $\mathbf{Q} = \Lambda^{-\frac{1}{2}}\mathbf{U}^T$. Λ is a diagonal matrix with the eigenvalues of matrix $\mathbf{R_x}$, where $\mathbf{R_x} = E(\mathbf{xx}^T)$

and $\mathbf{U}$ is a matrix with the corresponding eigenvectors of $\mathbf{R_x}$ as its columns. Matrix $\mathbf{Q}$ transforms the matrix $\mathbf{X}$ as

$$\mathbf{Z} = \mathbf{QX} = \mathbf{QAS} = \mathbf{BS}. \tag{2}$$

It is easy to show that $\mathbf{B}$ is the unitary matrix. Thus, the problem of finding a full rank matrix $\mathbf{A}$ with m^2 parameters is reduced to finding an orthonormal matrix $\mathbf{B}$ with $\frac{m(m-1)}{2}$ parameters. Equation (2) can be rewritten as $\mathbf{S} = \mathbf{B}^T\mathbf{Z}$ with two constraints: first, $\mathbf{B}$ is an orthonormal matrix and second, $s_i(k)$ as the i^{th} element of the k^{th} column, is statistically independent from the other elements of this column. Fast ICA (FICA), which employs the fixed point optimization algorithm, consists of two parts; in the first part, an objective function and an optimization algorithm are determined. In the second part, orthogonalization needs to be done in order to find m ICs. In the optimization algorithm, the projection method is used to optimize the objective function with constraint (Hyvarinen, Karhunen and Oja, 2001).

The fast ICA algorithm proceeds as follows (Lee, Yoo and Lee, 2004a,b):
Step 1: Set counter $i = 1$ and choose m.
Step 2: Choose the initial vector $\mathbf{b}_i$, it is a unit norm random vector ($\mathbf{b}_i$ is the i^{th} column of $\mathbf{B}$).
Step 3: Compute $\mathbf{b}_i \leftarrow E\left\{\mathbf{z}\dot{G}(\mathbf{b}_i{}^T\mathbf{z})\right\} - E\left\{\ddot{G}(\mathbf{b}_i{}^T\mathbf{z})\right\}\mathbf{b}_i$, where $\dot{G}$ and $\ddot{G}$ are the first and the second derivative of G. The function $G(s)$ is defined as $tanh(s)$ or $-\exp\left(\frac{-s^2}{2}\right)$ or s^4, etc. (see Hyvarinen, 1999, or Hyvarinen, Karhunen and Oja, 2001).
Step 4: Orthogonalize $\mathbf{b}_i$ using the Gram-Schmidt method as:

$$\mathbf{b}_i \leftarrow \mathbf{b}_i - \sum_{j=1}^{i-1}(\mathbf{b}_j^T\mathbf{b}_i)\mathbf{b}_j. \tag{3}$$

Step 5: Normalize $\mathbf{b}_i$ as

$$\mathbf{b}_i \leftarrow \frac{\mathbf{b}_i}{\|\mathbf{b}_i\|}. \tag{4}$$

Step 6: If $\mathbf{b}_i$ does not converge, go back to Step 3, else it is the i^{th} vector of $\mathbf{B}$. Then, if $i < m$, set $i = i + 1$ and go back to Step 2.
To summarize, the matrix $\mathbf{B}$ is estimated by the above mentioned method, and the IC matrix and the de-mixing matrix are calculated by Equations (5) and (6), respectively:

$$\mathbf{S} = \mathbf{B}^T\mathbf{Z} \tag{5}$$

$$\mathbf{W} = \mathbf{B}^T\mathbf{Q}. \tag{6}$$

The DICA-based fault detection algorithm has an offline part for training and an online part for testing (Lee, Yoo and Lee, 2004a, b). In other words, in the

offline part, the algorithm estimates a linear model of the system (matrix **A**) and estimates the source signals as the ICs matrix (matrix **S**). Then, the ICs matrix is divided into two parts in order to decrease the matrix dimensions. And in the last step, the algorithm calculates the 2-norm of each column of the ICs matrix to determine the control limit. In the online part, in each sample, the algorithm calculates the sources by the offline linear model and calculates the 2-norm of these source samples. A fault is detected in this sample if the 2-norms of ICs vectors are bigger than the control limit. The offline part of the algorithm is explained as follows:
Step 1: Construct the observation matrix $\mathbf{X} \in R^{m\times N}$ from the data. Then, choose the number of lag measurements ($l = 2$ is suggested), form the augmented matrix $\mathbf{X}_L \in R^{(l+1)m\times N}$ and normalize it (Lee, Yoo and Lee, 2004b).
Step 2: Whiten the $\mathbf{X_L}$ matrix using the whitening matrix (see equation (2)).
Step 3: Execute the ICA algorithm to calculate the de-mixing matrix **W** and then separate it by using the row norms of matrix **W** as:

$$\mathbf{W} = \begin{pmatrix} \mathbf{W_d} \\ \hline \mathbf{W_e} \end{pmatrix}, \tag{7}$$

where $\mathbf{W}_d$ corresponds to the bigger values of 2-norm of **W** rows, and $\mathbf{W}_e$ corresponds to smaller values of 2-norms of **W** rows.
Step 4: Calculate the ICs matrices, the dominant part, $\mathbf{S_d}$, and the excluded part, $\mathbf{S_e}$, of matrix **S** as:

$$\mathbf{s}_d(k) = \mathbf{W}_d\mathbf{x}(k) \quad k = 1, \ldots, N \tag{8}$$

$$\mathbf{s}_e(k) = \mathbf{W}_e\mathbf{x}(k) \quad k = 1, \ldots, N. \tag{9}$$

Step 5: Calculate three statistics to monitor the process:

- The systematic dominant part: $I^2{}_d(k) = \mathbf{s}_d{}^T(k)\mathbf{s}_d(k)$
- The systematic excluded part: $I^2{}_e(k) = \mathbf{s}_e{}^T(k)\mathbf{s}_e(k)$
- The non-systematic part: $SPE(k) = \mathbf{e}^T(k)\mathbf{e}(k)$

where $\mathbf{e}(k)$ and $\hat{\mathbf{x}}_L(k)$ are defined by Equations (10) and (11):

$$\mathbf{e}(k) = \mathbf{x}_L(k) - \hat{\mathbf{x}}_L(k) \tag{10}$$

$$\hat{\mathbf{x}}_L(k) = \mathbf{Q}^{-1}\mathbf{B}_d\mathbf{W}_d\mathbf{x}_L(k). \tag{11}$$

$\mathbf{B}_d$ is the dominant part of **B** (corresponding to $\mathbf{S}_d$), with $\mathbf{B} = [\mathbf{B}_d\mathbf{B}_e]$.
Step 6: Calculate the confidence limits of the three statistics by the kernel density estimation (KDE) (99% or 95%) to determine the control limit. KDE is a technique that appropriately determines these statistics as the control limits, while it simultaneously satisfies the confidence bounds. A univariate kernel estimator with kernel k is defined as $\hat{f}(x) = \frac{1}{Nh}\sum_{i=1}^{N} k(\frac{x-x_i}{h})$, where x is the considered data point, x_i is the observation, h is a constant value of the window width, and k is the kernel function that is usually chosen as the Gaussian density function (Lee, Yoo and Lee, 2004a).

The online part of the algorithm is explained as follows:
Step 1: Construct the new observation vector $\mathbf{x}_L(k)$ as in Step 1 of the offline part.
Step 2: Calculate the dominant and excluded parts of the ICs vector according to the Equations (8) and (9), provided above.

Step 3: Calculate three statistical indices as in Step 5 of the offline part. A fault has occurred if these new statistics are out of the control limit.

3. The proposed FDM and its advantages

3.1. The basic constructions

In the DICA-based FDM, which is called *the former FDM* in the rest of this paper, the dynamics of the system is modeled by constructing the augmented matrix $\mathbf{X}_L \in R^{(l+1)m\times(N+1)}$ from the observation matrix $\mathbf{X}_0 \in R^{m\times(N+l+1)}$ (see Equations (12) and (13) in Table 1). The disadvantage of this approach is that the dimensions of the augmented matrix increases dramatically.

Table 1: Construction of $\mathbf{X}_0, \mathbf{X}_L$ and $\mathbf{X}^{(0)}, \mathbf{X}^{(1)}, ..., \mathbf{X}^{(l)}$

$\mathbf{X}_0$ to be used in ICA	$$\mathbf{X}_0 = \left[\mathbf{x}(k)\mathbf{x}(k+1)...\mathbf{x}(k+N+l)\right] \quad (12)$$ where $\mathbf{x}(k)$ denotes the k^{th} vector of $\mathbf{X}_o$
$\mathbf{X}_L$ to be used in DICA	$$\mathbf{X}_L = \begin{bmatrix} \mathbf{x}(k) & \mathbf{x}(k+1) & \ldots & \mathbf{x}(k+N) \\ \mathbf{x}(k+1) & \mathbf{x}(k+2) & \ldots & \mathbf{x}(k+N+l) \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{x}(k+l) & \mathbf{x}(k+l+1) & \ldots & \mathbf{x}(k+l+N) \end{bmatrix} \quad (13)$$
Matrices $\mathbf{X}^{(0)}$, $\mathbf{X}^{(1)}, ..., \mathbf{X}^{(l)}$ to be used in the proposed ICA	$$\mathbf{X}^{(0)} = \left[\mathbf{x}(k)\mathbf{x}(k+1)\ldots\mathbf{x}(k+N)\right] \quad (14)$$ $$\mathbf{X}^{(1)} = \begin{bmatrix} \mathbf{x}(k+1) - \mathbf{x}(k), ..., \mathbf{x}(k+N+1) \\ -\mathbf{x}(k+N) \end{bmatrix} \quad (15)$$ ... $$\mathbf{X}^{(l)} = [\mathbf{x}^{(l-1)}(k+1) - \mathbf{x}^{(l-1)}(k), ..., \mathbf{x}^{(l-1)} (k+N+1) - \mathbf{x}^{(l-1)}(k+N)] \quad (16)$$ where $\mathbf{x}^{(l)}(k+j)$ is the $(j+1)^{th}$ column of matrix $\mathbf{X}^{(l)}$.

The proposed FDM constructs a set of $l+1$ matrices $(\mathbf{X}^{(0)}, \mathbf{X}^{(1)}, \mathbf{X}^{(2)}, ..., \mathbf{X}^{(l)})$ with m rows instead of a matrix $(\mathbf{X}_L)$ with $m(l+1)$ rows. The first matrix of this set is identical to N columns of the matrix $\mathbf{X}_o$, namely the observation matrix $\mathbf{X}^{(0)} \in R^{m\times N}$ (see Equation (14) in Table 1). The second matrix of this set is a first forward difference observation matrix $\mathbf{X}^{(1)} \in R^{m\times N}$, which is constructed as Equation (15) of Table 1, and so on. In general, the $(l+1)^{th}$ matrix of this set is the l^{th} forward difference observation matrix $\mathbf{X}^{(l)} \in R^{m\times N}$, with its elements being calculated by Equation (16) of Table 1. In $\mathbf{X}^{(i)}$, i is an index: this notation is used for other matrices and vectors in the rest of the paper. Actually, the proposed FDM checks the trend of each variable. In the continuous signal, this means that the proposed algorithm checks the values of the signal, the first derivative of the signal, the second derivative of the signal and so on.

After constructing $\mathbf{X}^{(0)}, \mathbf{X}^{(1)}, \mathbf{X}^{(2)}, ..., \mathbf{X}^{(l)}$, similarly as in Lee, Yoo and Lee (2004a), the ICA-based FDM (and not the DICA-based FDM) is executed for each $\mathbf{X}^{(0)}, \mathbf{X}^{(1)}, \mathbf{X}^{(2)}, ..., \mathbf{X}^{(l)}$ separately, and if any of them detects the fault, a fault has occurred.

It is worth mentioning that, in comparison with the former FDM, the proposed FDM constructs a low dimensional set of matrices. Therefore, it improves the detection capabilities by overcoming the drawbacks, related to dealing with high dimensional matrices.

3.2. The proposed algorithm

The proposed algorithm is summarized in two parts; the offline part and the online part, for training and testing, respectively. The offline part of the proposed method proceeds as follows:

Step 1: Choose the number of lag measurements (preferably $l = 2$) and construct the $l+1$ matrices $\mathbf{X}^{(0)}, ..., \mathbf{X}^{(l)}$ from $N+l+1$ samples of variables following the Equations (14), (15) and (16) of Table 1, and then normalize them.

Step 2: Whiten the $\mathbf{X}^{(0)}, ..., \mathbf{X}^{(l)}$, using the whitening matrices $\mathbf{Q}^{(0)}, ..., \mathbf{Q}^{(l)}$ to obtain $\mathbf{Z}^{(0)}, ..., \mathbf{Z}^{(l)}$, respectively, as:

$$\mathbf{Z}^{(i)} = \mathbf{Q}^{(i)}\mathbf{X}^{(i)} \tag{17}$$

where $\mathbf{Q}^{(i)} = \Lambda^{(i)\frac{-1}{2}}\mathbf{U}^{(i)^T}$, $\Lambda^{(i)}$ are diagonal matrices with the eigenvalues of matrices ${R_x}^{(i)}$. Here, ${\mathbf{R_x}}^{(i)} = E(\mathbf{x}^{(i)}\mathbf{x}^{(i)^T})$ and $\mathbf{U}^{(i)}$ are matrices with corresponding eigenvectors of ${\mathbf{R_x}}^{(i)}$ as the columns.

Step 3: Run the ICA algorithm for whitened matrices, to calculate the de-mixing matrices $\mathbf{W}^{(0)}, ..., \mathbf{W}^{(l)}$ (corresponding to $\mathbf{Z}^{(0)}, ..., \mathbf{Z}^{(l)}$), respectively. Then, calculate the IC vectors $\mathbf{s}^{(0)}(k), ..., \mathbf{s}^{(l)(k)}$ as

$$\mathbf{s}^{(i)}(k) = \mathbf{W}^{(i)}\mathbf{x}^{(i)}(k), \quad i = 0, ..., l \ \& \ k = 1, 2, ..., N. \tag{18}$$

Step 4: Separate each IC matrix by using the row norms of the de-mixing matrices, into two kinds of parts: the dominant parts and the excluded parts of matrices $\mathbf{S}^{(0)},...,\mathbf{S}^{(l)}$, corresponding to matrices $\mathbf{S}^{(0)}{}_d,...,\mathbf{S}^{(l)}{}_d$, and $\mathbf{S}^{(0)}{}_e,...,\mathbf{S}^{(l)}{}_e$, respectively.

Step 5: Calculate three groups of statistics for each of matrices $\mathbf{S}^{(0)},...,\mathbf{S}^{(l)}$, respectively, to monitor the process, as:

- The systematic dominant part ${I_d^{(i)}}^2(k) = {\mathbf{s}_d^{(i)}}^T(k)\mathbf{s}_d^{(i)}(k),\ i = 0,...,l$
- The systematic excluded part ${I_e^{(i)}}^2(k) = {\mathbf{s}_e^{(i)}}^T(k)\mathbf{s}_e^{(i)}(k),\ i = 0,...,l$
- The non-systematic part $SPE^{(i)}(k) = {\mathbf{e}^{(i)}}^T(k)\mathbf{e}^{(i)}(k),\ i = 0,...,l,$

where $\mathbf{e}^{(i)}(k)$ and $\hat{\mathbf{x}}^{(i)}(k)$ are calculated from Equations (19) and (20),

$$\mathbf{e}^{(i)}(k) = \mathbf{x}^{(i)} - \hat{\mathbf{x}}^{(i)}(k), \quad i = 0,...,l \tag{19}$$

$$\hat{\mathbf{x}}^{(i)}(k) = {\mathbf{Q}^{(i)}}^{-1}\mathbf{B}_d{}^{(i)}\mathbf{W}_d{}^{(i)}\mathbf{x}_d{}^{(i)}(k),\ i = 0,...,l. \tag{20}$$

$\mathbf{B}_d{}^{(0)},...,\mathbf{B}_d{}^{(l)}$ and $\mathbf{W}_d{}^{(0)},...,\mathbf{W}_d{}^{(l)}$ are the dominant parts of $\mathbf{B}^{(0)},...,\mathbf{B}^{(l)}$ and $\mathbf{W}^{(0)},...,\mathbf{W}^{(l)}$ (corresponding to matrices $\mathbf{S}_d{}^{(0)},...,\mathbf{S}_d{}^{(l)}$) as:

$$\begin{cases} \mathbf{W}^{(i)} = \begin{pmatrix} \mathbf{W}_d{}^{(i)} \\ \mathbf{W}_e{}^{(i)} \end{pmatrix} \\ \mathbf{B}^{(i)} = \begin{pmatrix} \mathbf{B}_d{}^{(i)}\mathbf{B}_e{}^{(i)} \end{pmatrix}, \quad i = 0,...,l. \end{cases} \tag{21}$$

Step 6: Calculate the confidence limits of the three groups of statistics for each of matrices $\mathbf{S}^{(0)},...,\mathbf{S}^{(l)}$, respectively, by KDE (99% or 95%) to determine the control limits.

The online part of the proposed method is realised as follows:

Step 1: Construct the new set of vectors $\mathbf{x}^{(0)}(k),...,\mathbf{x}^{(l)}(k)$ as in Step 1 of the offline part.

Step 2: Calculate the dominant and the excluded parts of the ICs set as:

$$\begin{cases} \mathbf{s}_d{}^{(i)}(k) = \mathbf{W}_d{}^{(i)}\mathbf{x}^{(i)}(k) \\ \mathbf{s}_e{}^{(i)}(k) = \mathbf{W}_e{}^{(i)}\mathbf{x}^{(i)}(k) \quad i = 0,...,l \end{cases}. \tag{22}$$

Step 3: Calculate three groups of the statistical indices as in Step 5 of the offline part. A fault has occurred if each of these new statistics is out of their control limits.

3.3. The advantages of the proposed FDM

In this subsection the advantages of the proposed FDM are explained, and in Section 4, they are validated by numerical results. The number of rows in each of the matrices $\mathbf{X}^{(i)} \in R^{m\times(N+1)}$ $i = 0,..,l$ (in the proposed FDM) is reduced in comparison to the matrix $\mathbf{X}_L \in R^{m(l+1)\times(N+1)}$ (in the former FDM) $(m < m(l{+}1))$. Thus, the proposed method improves the capabilities by overcoming the drawbacks of dealing with high dimensional matrices, which have been discussed already previously, including the advantages of robust performance, reduction of complexity, improvement of the sensitivity to whitening, and it can also constitute a strategy for dealing with the small sample size problem (see Section 1). The here mentioned advantages of the analytic proposed FDM are explained in the subsequent sections.

3.3.1. Improved running time

In practice, FDMs should not require powerful computational facilities as they are not available at many industrial sites. Thus, the FDMs that reduce the arithmetic operations and the running time are more practical.

In the ICA-based FDM, the offline part must be executed periodically, to re-model the system (Jelali, 2006). It can be seen that the running time of the offline part is significantly longer than that of the online part. Also in the offline part, the running time of the ICA step is significantly longer than the sum of the other steps of the FDM. In summary, the ICA running time is a rough estimation of the former FDM running time.

In the proposed FDM, the running time decreases in relation to the former FDMs. The reasons causing this are summarized in the following subsections.

3.3.1.1 Reduced number of arithmetic operations in k iterations

Reducing the number of arithmetic operations in k iterations is the main reason of reducing the running time. One iteration consists of steps 2 to 6 of the ICA algorithm, see Section 2. In this subsection, convergence of the i^{th} column of matrix $\mathbf{B}$ and each of the columns of matrices $\mathbf{B}^{(n)}$, $n = 0, ..., l$, in k^{th} iteration are assumed. In the former FDM, the number of arithmetic operations of steps 3, 4 and 5 of the ICA algorithm are shown byS_{f3}, S_{f4} and S_{f5}, respectively. In the proposed FDM, the number of arithmetic operations of steps 3, 4 and 5 of the ICA algorithm are expressed by S_{p3}, S_{p4} and S_{p5}, respectively. Lemmata 1 and 2 show an approximation of the ratios of $\frac{S_{f3}}{S_{p3}}$ and $\frac{S_{f4}}{S_{p4}}$ and Lemma 3 shows the ratio $\frac{S_{f5}}{S_{p5}}$.

LEMMA 1 *The ratio of* $\frac{S_{f3}}{S_{p3}}$ *of step 3* ($\boldsymbol{b}_i \leftarrow E\left\{\boldsymbol{z}\dot{G}({\boldsymbol{b}_i}^T\boldsymbol{z})\right\} - E\left\{\ddot{G}({\boldsymbol{b}_i}^T\boldsymbol{z})\right\}\boldsymbol{b}_i$) *is approximated by* $(l+1)^2$, *if each column of matrix* $\boldsymbol{B}$ *and each column of matrices* $\boldsymbol{B}^{(n)}$ $n = 0, 1, ..., l$ *converge in the* k^{th} *iteration.*

LEMMA 2 *The ratio of* $\frac{S_{f4}}{S_{p4}}$ *of step 4* ($\boldsymbol{b}_i \leftarrow \boldsymbol{b}_i - \sum_{j=1}^{i-1}({\boldsymbol{b}_i}^T\boldsymbol{b}_j)\boldsymbol{b}_j$) *is approximated by* $(l+1)^3$ *if each column of matrix* $\boldsymbol{B}$ *and each column of matrices* $\boldsymbol{B}^{(n)}$, $n = 0, 1, ..., l$ *converge in the* k^{th} *iteration.*

LEMMA 3 *The ratio of* $\frac{S_{f5}}{S_{p5}}$ *of step 5* ($\boldsymbol{b}_i \leftarrow \frac{\boldsymbol{b}_i}{\|\boldsymbol{b}_i\|}$) *calculated as* $(l+1)^2$ *if each column of matrix* $\boldsymbol{B}$ *and each column of matrices* $\boldsymbol{B}^{(n)}$, $n = 0, 1, ..., l$ *converge in the* k^{th} *iteration.*

It is worth mentioning that steps 2 and 6 do not involve high computational cost and the numbers of arithmetic operations of these steps are negligible. In practice, $l > 1$ is assumed and $l = 2$ is suggested by Lee, Yoo and Lee (2004b). Thus, in the former FDM, the number of arithmetic operations of steps 3, 4 and 5 are, respectively, about 9 times, 27 times and 9 times higher than those performed in the proposed FDM.

3.3.1.2 Reduced number of iterations

In subsection 3.3.1.1, convergence of the i^{th} column of matrix $\mathbf{B}$ and the i^{th} columns of matrices $\mathbf{B}^{(n)}$, $n = 0, 1, ..., l$, in the k^{th} iteration was assumed. This subsection explains how in most cases, it is estimated that this number of iterations (in the proposed method) is lower than that in the former methods. For the purposes of this paper, we give three ways of reasoning about it:

1. The ICA algorithm uses the projection method to minimize the cost function. This method solves the minimization problem with an unconstrained method (Newton method) (see step 3 of the ICA algorithm in Section 2), but after each iteration step, the solution $\mathbf{b}_i$, available at that time, is projected orthogonally onto the constraint set, to satisfy the constraints (see steps 4 and 5 of the ICA algorithm in Section 2). If $\mathbf{b}_i$ converges, it is taken as the i^{th} column of matrix $\mathbf{B}$, otherwise the iteration step is repeated and we go back to step 3 of the ICA algorithm until $\mathbf{b}_i$ converges (Hyvarinen, Karhunen and Oja, 2001). Therefore, the reduction of the number of possible directions of the $\mathbf{b}_i$ vector leads to a decrease in the number of iterations. As a result, if the dimension of the $\mathbf{b}_i$ vector decreases, the number of possible directions decreases, too, and the algorithm converges in the lower number of iterations. In the former FDM, the vector $\mathbf{b}_i \in R^{m(l+1)\times 1}$ was estimated by the ICA algorithm. On the other hand, in the proposed FDM, the set of vectors ${\mathbf{b}_i}^{(n)} \in R^{m\times 1}$ $n = 0, 1, ..., l$, is estimated by the ICA algorithm. Therefore, in the proposed FDM, the dimensions of vectors are lower in relation to the former FDM ($m < m(l+1)$). For instance, if we construct a two rows observation matrix and choose $l = 1$, the proposed FDM searches for vectors ${\mathbf{b}_i}^{(0)} \in R^{2\times 1}$ and ${\mathbf{b}_i}^{(1)} \in R^{2\times 1}$ in two dimensional space, while the former FDM searches for vectors $\mathbf{b}_i \in R^{4\times 1}$ in four dimensional space. The number of possible directions of the vector $\mathbf{b}_i$ decreases from 4 to 2 in the proposed FDM, although the ICA algorithm has been repeated twice. Thus, it is estimated that the number of iterations in the proposed method is lower than in the former methods.
2. In general, the ICA algorithm searches for $\frac{a(a-1)}{2}$ unknown elements in the orthonormal matrix $\mathbf{B} \in R^{a\times a}$ (Hyvarinen, Karhunen and Oja, 2001). Consequently, the former FDM searches for $\frac{m(l+1)(m(l+1)-1)}{2}$ unknown elements in matrix $\mathbf{B}$, while the proposed FDM determines $\frac{(l+1)m(m-1)}{2}$ unknown elements in all the matrices $\mathbf{B}^{(0)}, ..., \mathbf{B}^{(l)}$. However, the number of elements of the $\mathbf{b}_i \in R^{m(l+1)\times 1}$ vectors in the former FDM is equal to the number of elements of ${\mathbf{b}_i}^{(n)} \in R^{m\times 1}$, $n = 0, 1, ..., l$, vectors in the proposed FDM. It is clear that the number of elements that the ICA algorithm should estimate decreases in the proposed FDM ($\frac{m(l+1)(m(l+1)-1)}{2} > \frac{(l+1)m(m-1)}{2}$), without any change in the steps of the algorithm. For in-

stance, if we construct a two rows observation matrix and choose $l = 1$, the ICA of the proposed FDM searches for one unknown element in each of the orthogonal matrices $\mathbf{B}^{(0)} \in R^{2\times 2}$ and $\mathbf{B}^{(1)} \in R^{2\times 2}$, while the ICA of the former FDM searches for six unknown elements in the orthogonal matrix $\mathbf{B} \in R^{4\times 4}$. As a result, it is estimated that the number of iterations in the proposed method is lower than in the former methods.

3. In the former FDM, the matrix $\mathbf{B} \in R^{m(l+1)\times m(l+1)}$ is estimated to minimize the statistical dependence between the elements of the vector $\mathbf{s}(k) \in R^{m(l+1)\times 1}$. On the other hand, in the proposed FDM, $l + 1$ matrices $\mathbf{B}^{(n)} \in R^{m\times m}$, $n = 0, 1, ..., l$, are estimated to minimize the statistical dependence between the elements of the corresponding vectors $\mathbf{s}^{(n)}(k) \in R^{m\times 1}$, $n = 0, 1, ..., l$. Therefore, in the proposed FDM, minimizing the statistical dependence among small sets is required in comparison with to the former FDM ($m < m(l + 1)$). For instance, if $l = 1$ is chosen in a two row observation matrix, the ICA of the proposed FDM minimizes the statistical dependence between two elements of vector $\mathbf{s}^{(0)}(k) \in R^{2\times 1}$ (namely ${s_1}^{(0)}(k)$ and ${s_2}^{(0)}(k)$). In addition, it minimizes the statistical dependence between two elements of vector $\mathbf{s}^{(1)}(k) \in R^{2\times 1}$ (${s_1}^{(1)}(k)$ and ${s_2}^{(1)}(k)$), while there may exist a mutual statistical dependence between the elements of two vectors, such as, for instance, a dependence between ${\mathbf{s}_1}^{(0)}(k)$ and ${\mathbf{s}_1}^{(1)}(k)$ or ${\mathbf{s}_2}^{(1)}(k)$. On the other hand, the ICA of the former FDM minimizes the statistical dependence among four elements of the vector $\mathbf{s}(k) \in R^{4\times 1}(s_1(k),\ s_2(k),\ s_3(k)$ and $s_4(k))$. Thus, the number of iterations in the proposed method can be lower than in the former methods.

As indicated above, in most cases, it is estimated that the number of iterations in the proposed method is lower than in the former methods.

3.3.2. Reduced estimation error

In the ICA algorithm, the columns of the matrix are orthogonalized using the Gram-Schmidt orthogonalization method. In this method, the estimation errors of the first vectors are cumulated in the subsequent ones. Consequently, the cumulative error due to Gram-Schmidt orthogonalization method increases dramatically as the matrix dimensions increase (Hyvarinen, Karhunen and Oja, 2001). Thus, the estimation error is smaller in the proposed FDM in comparison with the former FDM, since the order of the matrices $\mathbf{B}^{(0)}, ..., \mathbf{B}^{(l)}$ in the proposed FDM is smaller than that of $\mathbf{B}$ in the former FDM ($m < m(l + 1)$).

3.3.3. Fault detection capabilities

In the former and the proposed FDM, fault detections occur when the 2-norm operators of ICs in the online part are out of the normal operating intervals (control limits). In certain circumstances, the former fault detection methods cannot detect the fault when $l = 1$ and the fault signal produces two terms of

ICs with opposite signs in different rows of an ICs vector. To illustrate this point, let us remind that when the former fault detection method constructs the augmented observation vectors, this leads to the higher order ICs vectors. In these higher order ICs vectors, half of the elements are the k^{th} sample of sum of source signals in each loop and the rest of the elements are $(k+1)^{th}$ sample of the sum of source signals in the same loops. Therefore, when a fault occurs and the DICA algorithm is executed, two elements, which are in different rows of the faulty IC vector are directly affected by the fault. One is equal to the simultaneous sum of two terms (a term corresponding to the source signals, and another term, corresponding to the fault source signal in one loop) in the k^{th} sample, while the other one equals to the same summation only for the $(k+1)^{th}$ sample. As a result, the sum of squares of these two elements of the faulty IC vector, when two consecutive samples of the fault source signal have opposite signs, can cancel out each other. This leads to failure in fault detection because the sum of the squares of the faulty IC vector elements has not exceeded the limits of the normal operating interval (the control limit). For instance, a system has been run with two interfering loops. The observation matrix has been constructed by taking the samples from this system and the ICA algorithm has been executed to estimate the source signals. Assume that the ICs vectors $\mathbf{s}^{(0)}(k)$ and $\mathbf{s}^{(0)}(k+1)$ (corresponding to observation matrix $\mathbf{X}^{(0)}$) have been calculated as (-1,1) and (1,1), respectively, by the ICA in the offline part of the proposed FDM, while in the offline part of the former FDM, the ICs vectors $\mathbf{s}(k)$ and $\mathbf{s}(k+1)$ (corresponding to the augmented matrix $\mathbf{X}_L$) have been calculated as (-1, 1, 1, 1) and (1, 1, 1, -1), respectively, by the ICA algorithm. Thus, in the proposed FDM, the points (-1, 1) and (1, 1) are the normal operating data, and the normal operating interval in these points is calculated as $((-1)^2+1^2=2)$. At the same time, in the former FDM, the points (-1, 1, 1, 1) and (1, 1, 1, -1) are the normal operating data, and the normal operating interval, corresponding to them, is calculated as $((-1)^2+1^2+1^2+1^2=4)$.

In the online part, the system is run again and a fault occurs. The ICs corresponding to the fault are 0.2 in sample k and -0.3 in sample $k+1$. Thus, the ICs vector $\mathbf{s}(k)$ changes from $(-1,1,1,1)$ to $(-1,1+0.2,1,1-0.3)$ in the former FDM. As a result, the fault detection statistic is equal to 3.93 $((-1)^1+1.2+1^2+0.7^2=3.93)$. This result does not violate the fault detection criteria $(3.93<4)$ and hence the fault remains undetected. The reason is that two consecutive samples of fault cancel each other in the former FDM. In the proposed FDM, on the other hand, the dimensions of the ICs vector decrease and two consecutive samples of fault are not located in one vector. In other words, the ICs vectors $s(k)$ and $s(k+1)$ change from $(-1,1)$ and $(1,1)$ to $(-1,1+0.2)$ and $(1,1-0.3)$. Therefore, the fault detection statistic in the k^{th} sample of the online part is equal to 2.44 $(1^2+1.2^2=2.44)$. Therefore, the fault is detected, because the fault detection statistic is bigger than the control limit $(2.44>2)$. It can be easily generalized to the case of $l>1$. In this case, if three or more consecutive samples of the fault source signal have opposite signs, the sum of squares of the faulty IC vector might cancel out each other and the former FDM fails to detect the fault.

On the other hand, although, as mentioned above, the proposed FDM has definite advantages, it increases the number of false alarms, too.

4. Simulation results

To validate the algorithm, the former and the proposed FDMs are applied to two systems:

4.1. The coupled tank system

A simple nonlinear model of the coupled tank has been provided in Ko, Chen, Chen, Zhuang and Chen Tan (2001) as:

$$\frac{dH_1}{dt} = \frac{Q_{i1}}{A_1} - \frac{\alpha_1}{A_1}\sqrt{H_1} - \frac{\alpha_3}{A_1} sgn(H_1 - H_2)\sqrt{H_1 - H_2} \quad (23)$$

$$\frac{dH_2}{dt} = \frac{Q_{i2}}{A_2} - \frac{\alpha_2}{A_2}\sqrt{H_2} - \frac{\alpha_3}{A_2} sgn(H_2 - H_1)\sqrt{H_1 - H_2} \quad (24)$$

where H_1, H_2 are the heights of fluid in tanks 1 and 2, respectively; Q_{i1} and Q_{i2} are the pump flow rates into tanks 1 and 2, respectively; α_1, α_2 and α_3 are constants; A_1, A_2 are the cross sectional areas of tanks 1 and 2, respectively. Equations (23) and (24) form a system with two interfering loops, thus the fault is propagated through the loops. In order to decrease the error signals, a lag controller is designed for both the upper (tank 1) and lower (tank 2) loop; $G_{C1} = \frac{s+0.9}{s+0.0002}$ and $G_{C2} = \frac{s+0.98}{s+0.001}$ (see Fig. 1). Two sinusoidal signals are added to output signals in modeling of the waves on the surface. The amplitudes of the sinusoidal signals are 0.03 m and 0.08 m and their frequencies are 0.2 Hz and 0.5 Hz in the loops of tank 1 and tank 2, respectively. A white noise with the power of about 0.002 is added to each loop. Other parameters of Equations (23) and (24) are set as $\alpha_1 = 10.78$, $\alpha_2 = 11.03$, $\alpha_3 = 11.03$, $A_1 = 3m^2$, $A_2 = 32m^2$, $H_1 = 17m$, and $H_2 = 15m$, see Fig. 1.

The simulation runs for 250 seconds and 5000 samples are accumulated. The first 1500 samples are neglected, because they correspond to the transient response and the last 500 samples are excluded when establishing the augmented observation matrix. Thus, the 3000 samples are accumulated as the dataset; around 750 samples are treated as training data and 2250 samples as testing data.

A ramp type of level sensor fault is added to the control loop of tank 2, between samples 1400 to 1600 (see Fig. 2 (f)). For both the former and the proposed FDMs $l = 1$ is chosen. To calculate the confidence bound, KDE (99%) with Gaussian kernel is used. The simulation is repeated 100 times and the obtained fault detection and false alarm rates are shown in Table 2. It is clear that the proposed FDM improves the detection rate in comparison with the former method (although it increases the false alarm rate by about 1%, too). For instance, Fig. 2 shows one of these improved detection cases. Fig. 2(a) shows

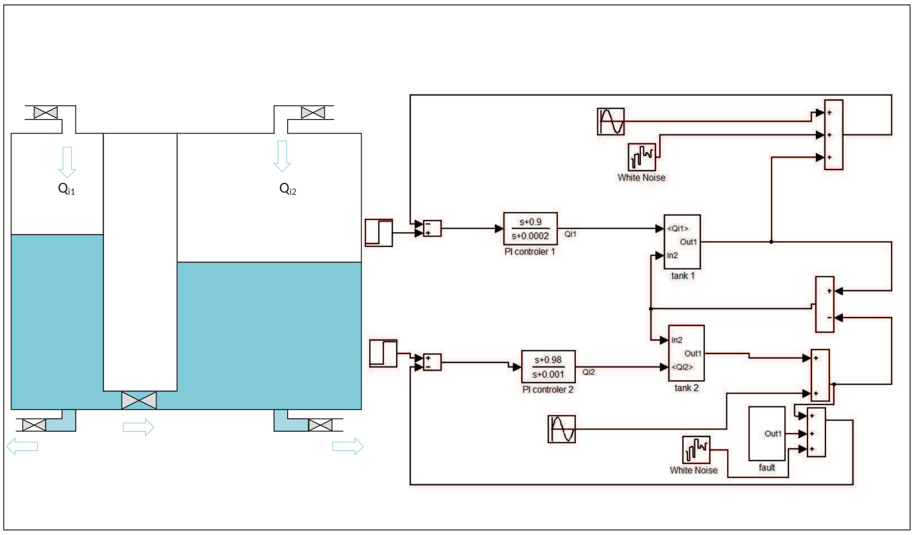

Figure 1: The coupled tanks system and its block diagram

Table 2: Fault detection rate and false alarm rate in the coupled tank system case

Sensor fault	False alarm rate by		Detection rate by	
	Proposed FDM	Former FDM	Proposed FDM	Former FDM
Step type	3%	2%	91%	87%

the fault detection statistics, ${I_d}^2(k)$ $k = 1, ..., N$, in the online and the offline parts, obtained from the former FDM. It is clear that the square of the 2-norm value of the faulty IC vectors between samples 1400 to 1600, do not exceed the normal operating interval limits, shown by the dashed line, in the former fault detection method. This occurs, because the two consecutive samples of the fault source signal have opposite signs and two ICs, corresponding to them, cancel out each other and the fault remains hidden, see subsection 3.3.3. Figs. 2(b) and 2(c) show the fault detection statistics, ${I_d}^{(0)^2}(k), {I_d}^{(1)^2}(k)$, $k = 1, ..., N$, respectively, in the online part and in the offline part. In Fig. 2 (c) the square of 2-norm value of the faulty IC vectors, corresponding to ${I_d}^{(1)^2}$, exceeds the normal operating interval limits, and so, the fault is being detected. At the same time, the square of 2-norm value of the faulty IC vectors, corresponding to ${I_d}^{(0)^2}$, does not exceed the normal operating interval limits, and cannot detect the fault, see Fig. 2(b). According to the proposed algorithm, fault will be detected if any of these statistics detects the fault. Consequently, the proposed FDM detects the fault, while the former FDM cannot. The former FDM cannot detect the fault, because two consecutive samples of fault source signal have opposite signs and the sum of squares of faulty IC vector, cancel out each other. This is the numerical validation of the considerations from the subsection 3.3.3.

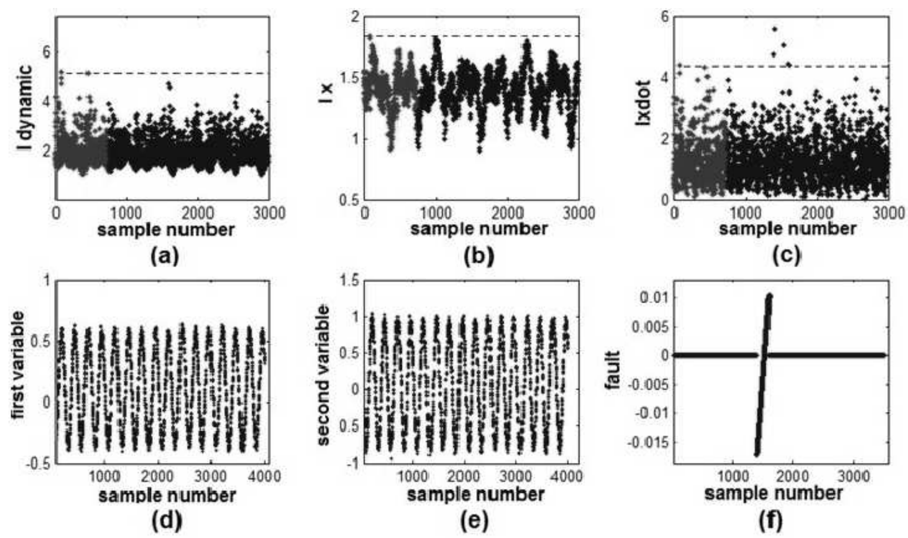

Figure 2: Comparison between the proposed and the former FDMs: (a) indicates how fault detection statistics cannot detect the fault in the former FDM; (c) shows how fault detection statistics ${I_d^{(1)}}^2$ detects the fault, while (b) shows that ${I_d^{(0)}}^2$ cannot detect the fault; (d) and (e) indicate the error signals that are sampled from the system; (f) shows the fault signal

The reductions in running time and number of iterations are the main advantages of the proposed FDM. Table 3 shows the running time and the number of iterations in the proposed and the former FDM. It indicates that the running time of ICA (2.0593 s) is a rough estimation of the total running time in the former FDM (2.107 s). Furthermore, Table 3 shows that the proposed FDM reduces the running time of the ICA step, while increasing the running time of the other steps. However, since the ICA step contributes significantly to the total running time, the total running time of the proposed FDM is significantly shorter than the total running time of the former FDM (2.107 >> 0.1156). This constitutes the numerical validation of the reasoning from subsection 3.3.1. Furthermore, Table 3 shows that the mean values of the numbers of iterations in the proposed FDM are smaller than in the former FDM (5042 >> 45.4). It is worth mentioning that the number of iterations in the proposed FDM is the sum of iterations, which are used to estimate ${\mathbf{b}_i}^{(0)}$ and ${\mathbf{b}_i}^{(1)}$. This, in turn, is the numerical validation of the reasoning from subsection 3.3.1.2.

To summarize, the proposed FDM decreases the running time and the number of iterations in comparison to the former FDM. According to the simulations, the running time and the number of iterations are significantly reduced, by 94% and 99%, respectively, in the proposed FDM in comparison with the former one.

Table 3: Mean values of the running time and the number of iterations in the proposed and former FDM

	Running time		Number of iterations	
	Proposed FDM	Former FDM	Proposed FDM	Former FDM
Construction of the observation matrix and whitening (Steps 1 and 2, offline)	0.0325 s	0.0249 s	—	—
Execution of ICA (Step 3, offline)	0.0471 s	2.0593 s	42.4	5042
Seperation of the matrices, calculation of statistics and control limits (Steps 4, 5 and 6, offline)	0.036 s	0.021 s	—	—

4.2. The Tennessee Eastman process

The Tennessee Eastman (TE) process is widely used to compare the performance of different FDMs. The simulation model and the flow sheet of TE process is introduced in http://depts.washington.edu and in Lee, Yoo and Lee (2004b), respectively. The major unit operations are a reactor, a condenser, a compressor, a separator and a stripper. The details and equations of the TE process are available in Jockenhovel, Biegler and Wachter (2003). The 21 process faults are introduced in Table 4. Similarly as in Lee, Yoo and Lee (2004b), 33 monitored process variables are used; they include 22 process measurements and 11 manipulated variables. The simulation is run for 3 minutes and 480 and 960 measurements are used as training and testing datasets, respectively; $l = 2$ is chosen and confidence bound is determined by KDE (99%) with Gaussian kernel.

The detection rates of the proposed FDM and the former FDM are shown in Table 4 (using fault detection statistics I^2 and AO, see Chun-Chin, Mu-Chen and Long-Sheng, 2010). The fault detection ability of the proposed method is clear. These results constitute the numerical validation of the considerations from the subsection 3.2.3.

Table 5 shows an obvious advantage of the proposed method in terms of running time and number of iterations ($20.741\ s < 100.48\ s$ and $1370 < 3360$).

According to the simulations, the running time and the number of iterations are considerably reduced, by 79% and 59%, respectively, in the proposed FDM in comparison with the former one. Thereby, we obtain the numerical validation of the reasoning from subsection 3.3.1.

5. Conclusions

A modified ICA-based FDM is introduced in this paper. The idea of the proposed FDM is to construct the observation matrix (not the augmented observation matrix), the first forward difference observation matrix, the second forward difference observation matrix, and so on, which reveals trends between the variable samples and the model of the dynamics of the system. In comparison to former FDM, the proposed FDM constructs low dimensional matrices. Therefore, it improves the detection capabilities by overcoming the drawbacks of dealing with high dimensional systems, which have been discussed in previous studies, related to such issues as robust performance, reduced analysis complexity, or the sensitivity to whitening, and it can constitute a strategy for dealing with the small sample size problem. Furthermore, compared to the former FDM, it is shown that the proposed FDM reduces the running time, since it reduces the number of iterations and arithmetic operations per iteration. Furthermore, the proposed FDM decreases the orthogonalization error, because it reduces the order of matrix **B**. Moreover, the proposed FDM is detecting a class of faults that the former fault detection method cannot detect. This class of faults occurs when two or more consecutive samples of the fault source signal have opposite signs and cancel out each other. Simulation results are used to verify the effectiveness of the proposed FDM.

6. Appendix A

There is a nested loop in the ICA algorithm, meant to estimate the matrix **B**. The inner loop estimates the i^{th} column of matrix **B**, denoted $\mathbf{b}_i$. When $\mathbf{b}_i$ converges, the outer loop shifts this action to the next column. Suppose k iterations should be performed for the i^{th} column of matrix **B** to converge and assume that each column of the matrices $\mathbf{B}^{(n)}, n = 0, 1, ..., l$, converges in k^{th} iteration, too. The approximation of the ratios of $\frac{S_{f3}}{S_{p3}}$, $\frac{S_{f4}}{S_{p4}}$ and $\frac{S_{f5}}{S_{p5}}$ are calculated as in Lemmata 1, 2 and 3, respectively.

LEMMA 1. *The ratio of* $\frac{S_{f3}}{S_{p3}}$ *in step 3 (*$\boldsymbol{b}_i \leftarrow E\left\{\boldsymbol{z}\dot{G}(\boldsymbol{b}_i{}^T\boldsymbol{z})\right\} - E\left\{\ddot{G}(\boldsymbol{b}_i{}^T\boldsymbol{z})\right\}\boldsymbol{b}_i$*) is approximated by* $(l+1)^2$ *if each column of matrix* $\boldsymbol{B}$ *and each column of matrices* $\boldsymbol{B}^{(n)},\ n = 0, 1, ..., l$*, converge in the* k^{th} *iteration.*

PROOF. The number of arithmetic operations of step 3 for the i^{th} column of matrix **B** in the former FDM, and for each column of matrices $\mathbf{B}^{(n)}$, $n = 0, 1, ..., l$, in the proposed FDM, are shown in Table 6.

Table 4: The faults of TE process and detection rates (%)

Fault ID	Description	Type	Detection rate			
			Proposed FDM %		Former FDM%	
			I^2	AO	I^2	AO
IDV1	A/C feed ratio, B competition constant (stream 4)	Step	100	100	100	100
IDV2	B composition, A/C ratio constant (stream 4)	Step	99	100	99	99
IDV3	D feed temperature (stream 2)	Step	2	3	2	2
IDV4	Reactor cooling water inlet temperature	Step	100	100	99	100
IDV5	Condenser cooling water inlet temperature	Step	100	100	100	100
IDV6	A feed loss (stream 1)	Step	100	100	100	100
IDV7	C header pressure loss-reduced availability (stream 4)	Step	100	100	100	100
IDV8	A, B, C feed composition (stream 4)	Random	98	99	98	98
IDV9	D feed temperature (stream 2)	Random	3	4	2	3
IDV10	C feed temperature (stream 4)	Random	87	89	85	89
IDV11	Reactor cooling water inlet temperature	Random	65	82	63	83
IDV12	Condenser cooling water inlet temperature	Random	100	100	100	100
IDV13	Reaction kinetics	Slow drift	95	95	95	96
IDV14	Reactor cooling water valve	sticking	100	100	100	100
IDV15	Condenser cooling water valve	sticking	5	5	5	5
IDV16	Unknown	Unknown	86	92	84	92
IDV17	Unknown	Unknown	89	96	89	96
IDV18	Unknown	Unknown	91	91	90	91
IDV19	Unknown	Unknown	85	96	85	94
IDV20	Unknown	Unknown	89	92	88	92
IDV21	Valve position constant (stream 4)	Constant	47	63	47	62

Table 5: Mean values of the running time and the number of iterations in the proposed and former FDM

	Running time		Number of iteration	
	Proposed FDM	Former FDM	Proposed FDM	Former FDM
Construction of the observation matrix and whitening (Steps 1 and 2 - offline)	0.961 s	1.0604 s	—	—
Execution of ICA (step 3 - offline)	17.92 s	98.2 s	1370	3360
Separation of the matrices, Calculation statistics and control limits (steps 4, 5 and 6 - offline)	1.86 s	1.22 s	—	—

Table 6: The number of arithmetic operations of step 3 for the i^{th} column of matrix $\mathbf{B}$ in the former FDM and each column of matrices $\mathbf{B}^{(n)}$ in the proposed FDM

Proposed FDM		Former FDM	
Column	Number of arithmetic operations	Column	Number of arithmetic operations
$\mathbf{b}_i{}^{(0)}$	$\left[(N+1)(m^2+1)+2m\right]k$	$\mathbf{b}_i$	$[(N+1)\left[m^2(l+1)^2+1\right]+2m(l+1)]k$
$\mathbf{b}_i{}^{(1)}$	$\left[(N+1)(m^2+1)+2m\right]k$		
. . .	. . .		
$\mathbf{b}_i{}^{(l)}$	$\left[(N+1)(m^2+1)+2m\right]k$		
Total number of arithmetic operations	$S_{p3i}=(l+1)\left[(N+1)(m^2+1)+2m\right]k$	Total number of arithmetic operations	$S_{f3i}=[(N+1)\left[m^2(l+1)^2+1\right]+2m(l+1)]k$

The ratio of the total number of arithmetic operations of step 3 for the i^{th} column of matrix $\mathbf{B}$ in the former FDM (S_{f3i}) to that for each column of matrices $\mathbf{B}^{(n)},\ n=0,1,...,l$, in the proposed FDM ($S_{p3i}$) is calculated as:

$$\frac{S_{f3i}}{S_{p3i}}=\frac{\left[(N+1)\left[m^2(l+1)^2+1\right]+2m(l+1)\right]k}{(l+1)\left[(N+1)(m^2+1)+2m\right]k}. \tag{25}$$

In practice, N and m are large numbers, thus after straightforward algebraic simplification and upon ignoring the low values, this ratio is approximated as:

$$\frac{S_{f3i}}{S_{p3i}} \approx l+1. \tag{26}$$

Equation (26) shows that the total number of arithmetic operations of step 3 for each column of matrix $\mathbf{B}$ in the former FDM is $l+1$ times higher than in the proposed FDM. Furthermore, in the former FDM, $m(l+1)$ columns are determined by the outer loop, while in the proposed FDM, by contrast, m columns are determined by the outer loop. Therefore:

$$\frac{S_{f3i}}{S_{p3i}} \approx (l+1)^2. \tag{27}$$

□

Lemma 2. *The ratio of* $\frac{S_{f4}}{S_{p4}}$ *in step 4* $(\boldsymbol{b}_i \leftarrow \boldsymbol{b}_i - \sum_{j=1}^{i-1} ({\boldsymbol{b}_i}^T \boldsymbol{b}_j)\boldsymbol{b}_j)$ *is approximated by* $(l+1)^3$ *if each column of matrix* $\boldsymbol{B}$ *and each column of matrices* $\boldsymbol{B}^{(n)}$, $n = 0,1,...,l$, *converge in the* k^{th} *iteration.*

Proof. The number of arithmetic operations of step 4 for the i^{th} column of matrix $\mathbf{B}$ in the former FDM, and for each column of matrices $\mathbf{B}^{(n)}$, $n = 0,1,...,l$, in the proposed FDM, are shown in Table 7. In the proposed FDM,

Table 7: The number of arithmetic operations of step 4 for the i^{th} column of matrix $\mathbf{B}$ in the former FDM and each column of matrices $\mathbf{B}^{(n)}$ in the proposed FDM

Proposed FDM		Former FDM	
Column	Number of arithmetic operations	Column	Number of arithmetic operations
${\mathbf{b}_i}^{(0)}$	$\left[m^2(i-1)+m\right]k$	$\mathbf{b}_i$	$[m^2(l+1)^2(i-1)$ $+m(l+1)]k$
${\mathbf{b}_i}^{(1)}$	$\left[m^2(i-1)+m\right]k$		
. . .	. . .		
${\mathbf{b}_i}^{(l)}$	$\left[m^2(i-1)+m\right]k$		
Total number of arithmetic operations	$S_{p4i} = (l+1)$ $\left[m^2(i-1)+m\right]k$	Total number of arithmetic operations	$S_{f4i} = [m^2(l+1)^2$ $(i-1)+m(l+1)]k$

S_{p4i} shows the total number of arithmetic operations of ${\mathbf{b}_i}^{(n)}$, $n = 0,1,...,l$ (if each one of ${\mathbf{b}_i}^{(n)}$ converges in the k^{th} iteration). The outer loops increase the value of index i from 1 to m for determining the matrices $\mathbf{B}^{(n)}$, $n = 0,1,...,l$. Thus, the total number of arithmetic operations of step 4 in the proposed FDM is calculated as (S_{p4}):

$$S_{p4} = \sum_{i=1}^{m} (l+1) \left[m^2(i-1) + m \right] k = mk(l+1)(1 + \frac{m^2(m-1)}{2}). \tag{28}$$

In the former FDM, S_{f4i} shows the total number of arithmetic operations of $\mathbf{b}_i$ (if $\mathbf{b}_i$ converges in k^{th} iteration). The outer loop increases i from 1 to $m(l+1)$ in determination of the matrix $\mathbf{B}$. Thus, the total number of arithmetic operations of step 4 in the former FDM is calculated as (S_{f4}):

$$\begin{aligned} S_{f4} &= \sum_{i=1}^{m(l+1)} \left[m^2(l+1)^2(i-1) + m(l+1) \right] k \\ &= mk(l+1)(1 + \frac{m^2(l+1)^2(m(l+1)-1)}{2}). \end{aligned} \tag{29}$$

In practice, m is a large number, thus, after straightforward algebraic simplification and ignoring the low values, the ratio of $\frac{S_{f4}}{S_{p4}}$ can be approximated as:

$$\frac{S_{f4}}{S_{p4}} \approx (l+1)^3. \tag{30}$$

□

Lemma 3. *The ratio of* $\frac{S_{f5}}{S_{p5}}$ *in step 5 (* $\boldsymbol{b}_i \leftarrow \frac{\boldsymbol{b}_i}{\|\boldsymbol{b}_i\|}$*) is calculated as* $(l+1)^2$ *if each column of matrix* $\boldsymbol{B}$ *and each column of matrices* $\boldsymbol{B}^{(n)}$*,* $n = 0, 1, ..., l$*, converge in the* k^{th} *iteration.*

Proof. The number of arithmetic operations of step 5 for the i^{th} column of matrix $\mathbf{B}$ in the former FDM, and for each column of matrices $\mathbf{B}^{(n)}$, $n = 0, 1, ..., l$, in the proposed FDM, are shown in Table 8.

In the proposed FDM, S_{p5i} is the total number of arithmetic operations of ${\mathbf{b}_i}^{(n)}$, $n = 0, 1, ..., l$ (if each one of ${\mathbf{b}_i}^{(n)}$ converges in the k^{th} iteration). The outer loops increase the value of index i from 1 to m so as to determine the matrices $\mathbf{B}^{(n)}$, $n = 0, 1, ..., l$. Thus, the total number of arithmetic operations of step 5 in the proposed FDM is calculated as (S_{p5}):

$$S_{p5} = mS_{p5i} = (l+1)m^3k. \tag{31}$$

In the former FDM, S_{p5i} shows the total number of arithmetic operations related to $\mathbf{b}_i$ (if $\mathbf{b}_i$ converges in the k^{th} iteration). The outer loop increases the value of index i from 1 to $(l+1)$ so as to determine the matrix $\mathbf{B}$. Thus, the total number of arithmetic operations of step 5 in the former FDM is calculated as (S_{f5}):

$$S_{f5} = mS_{f5i} = m^3(l+1)^3k. \tag{32}$$

The ratio of $\frac{S_{f5}}{S_{p5}}$ is therefore calculated as:

$$\frac{S_{f5}}{S_{p5}} = (l+1)^2. \tag{33}$$

Table 8: The number of arithmetic operations of step 5 for the i^{th} column of matrix $\mathbf{B}$ in the former FDM and each column of matrices $\mathbf{B}^{(n)}$ in the proposed FDM

Proposed FDM		Former FDM	
Column	Number of arithmetic operations	Column	Number of arithmetic operations
${\mathbf{b}_i}^{(0)}$	m^2k	$\mathbf{b}_i$	$m^2(l+1)^2k$
${\mathbf{b}_i}^{(1)}$	m^2k		
. . .	. . .		
${\mathbf{b}_i}^{(l)}$	m^2k		
Total number of arithmetic operations	$S_{p5i} = (l+1)m^2k$	Total number of arithmetic operations	$S_{f5i} = m^2(l+1)^2k$

□

References

Chun-Chin H., Mu-Chen C., and Long-Sheng C. (2010) A Novel Process Monitoring Approach with Dynamic Independent Component Analysis. *Control Engineering Practice*, **18**(3), 242 – 253.

Deng W., Liu Y, Hu J. and Guo J. (2012) The small sample size problem of ICA: A comparative study and analysis. *Proc. on Pattern Recognition* **45**(12), 4438 – 4450.

Ge Z., Song Z. (2007) Process Monitoring Based on Independent Component Analysis Principal Component Analysis (ICA-PCA) and Similarity Factors. *Industrial and Engineering Chemistry Research*, **46**(7), 2054 – 2063.

Hehe M., Yi H. and Hongbo S. (2011) Statistics Kernel Principal Component Analysis for nonlinear process fault detection. *Intelligent Control and Automation (WCICA)*, 2011, 9th World Congress. IEEE Press, 431–436.

http://depts.washington.edu/control/larrry/TE/download.html

Hyvarinen A. (1999) Fast and Robust Fixed-Point Algorithms for Independent Component Analysis. *IEEE Trans. on Neural Networks*, **10**(3), 626 – 634.

Hyvarinen A., Karhunen J. and Oja E. (2001) *Independent Component Analysis*. John Wiley & Sons Inc., New York.

Isermann R. (2006) *Fault Diagnosis Systems. An Introduction From Fault Detection to Fault Tolerance*. Springer, Berlin.

JELALI M. (2006) An overview of control performance assessment technology and industrial applications. *Control Engineering Practice*, **14**(5), 441 – 466.

JOCKENHOVEL T., BIEGLER L. T., AND WACHTER A. (2003) Dynamic optimization of Tennessee Eastman Process using the OptControlCenter. *Computer and Chemical Engineering*, 27, 1513 – 1531.

KO C. C., CHEN B. M., CHEN J., ZHUANG Y., AND CHEN TAN K. (2001) Development of a Web-Based Laboratory for Control Experiments on a Coupled Tank Apparatus. *IEEE Trans. on Education*, **44**(1), 76 – 86.

KU V., STORER R. H., AND GEORGAKIS C. (1995) Disturbance detection and isolation by dynamic principal component analysis. *Chemometrics and Intelligent Laboratory System*, **30**(1), 179 – 196.

LE Q.V., KARPENKO A., NGIAM J. AND NG A.Y.(2011) ICA with Reconstruction Cost for Efficient Overcomplete Feature Learning. *Advances in Neural Information Processing Systems* 24, 1017 – 1025.

LEE T. W. (1998) *Independent Component Analysis. Theory and Applications*, Springer.

LEE J. M., YOO C. K., AND LEE I. B. (2004a) Statistical monitoring of dynamic processes based on dynamic independent component analysis. *Chemical Engineering Science*, **59**(14), 2995 – 3006.

LEE J. M., YOO C. K., AND LEE I. B. (2004b) Statistical Process Monitoring With Independent Component Analysis. *J. Process Control*, **14**(5), 467 – 485.

LEE J. M., QIN J., AND LEE I. B. (2006) Fault Detection and Diagnosis Based on Modified Independent Component Analysis. *AIChE Journal*, **52**(10), 3501 – 3514.

MARTIN E. B., AND MORRIS A. J. (1996) Non-Parametric Confidence Bounds for Process Performance Monitoring Charts. *J. Process Control*, **6**(6), 349 – 358.

NGUYEN V.H. AND GOLINVAL J.C. (2010) Fault detection based on Kernel Principal Component Analysis. *Engineering Structure*, **32**(11), 3683 – 369.

PUNTONET C. G. AND PRIETO A. (2004) *Independent Component Analysis and Blind Signal Separation*. Springer.

THORNHILL N. F., AND HORCH A. (2007) Advances and new directions in plant-wide disturbance detection and diagnosis. *Control Engineering Practice*, **15**(10), 1196 – 1206.

THARRAULT Y., MOUROT G., RAGOT J. AND MAQUIN D. (2008) Fault detection and isolation with robust principal component analysis. *International Journal of Applied Mathematics and Computer Science*, **18**(4), 429 – 442.

TIAN Y., DU W. AND QIAN F. (2013) Fault Detection and Diagnosis For Non-Gaussian Processes with Periodic Disturbance Based on $ARMA - ICA$. *Industrial and Engineering Chemistry Research*, **52** (34), 12082 – 12107.

VINZI V. E., CHIN W.W., HENSELER J. AND WANG H. (2010) *Handbook of Partial Least Squares Concept. Methods and Applications*. Springer, Berlin.

Wang L., Shi H. (2014) Improved Kernel PLS-Based Fault Detection Approach for Nonlinear Chemical Processes. *Chinese Journal of Chemical Engineering*, **22**(6), $657 - 663$.

Zhang Yw. and Zhang Y. (2010) Fault detection of non-Gaussian processes based on modified independent component analysis. *Chemical Engineering Science*, **65**(16), $4630 - 4639$.

Zhao X., Xue Y. and Wang T. (2014) Fault detection of batch process based on multiway Kernel $T - PLS$. *Journal of Chemical and Pharmaceutical Research*, **6**(7), $338 - 346$.

Zhao X. and Xue Y. (2014) Output Relevant Fault Detection and Identification of Chemical Process Based on Hybrid Kernel $T - PLS$. *The Canadian Journal of Chemical Engineering*, **92**(10), $1822 - 1828$.