Penalized reconstruction-based multivariate contribution analysis for fault isolation

Bo He\textsuperscript{a,b}, Jie Zhang\textsuperscript{b,*}, Tao Chen\textsuperscript{c}, Xianhui Yang\textsuperscript{a}

\textsuperscript{a} Department of Automation, Tsinghua University, Beijing 100084, China
\textsuperscript{b} School of Chemical Engineering and Advanced Materials, Newcastle University, Newcastle upon Tyne NE1 7RU, UK
\textsuperscript{c} Department of Chemical and Process Engineering, University of Surrey, Guildford GU2 7XH, UK

* Corresponding author; Tel.: +44-191-2227240; Email: jie.zhang@newcastle.ac.uk

Abstract – Contribution analysis in multivariate statistical process monitoring (MSPM) identifies the most responsible variables to the detected process fault. In multivariate contribution analysis, the main challenge of fault isolation is to determine the appropriate variables to be analysed and this usually results in a combinatorial optimisation problem. Reconstruction-based multivariate contribution analysis (RBMCA) is a generic framework to solve this problem. This paper derives a sufficient condition for the isolatability of faulty variables when using RBMCA. In addition, a penalized RBMCA (PRBMCA) framework is developed to enhance the effectiveness and efficiency of fault isolation, especially for process faults with small magnitude. In contrast to the original RBMCA, this penalized solution includes two steps. L\textsubscript{1} – penalized reconstruction is used in the first step to obtain a more compact set of faulty variables. Then, the original RBMCA with branch and bound algorithm is implemented to further narrow down the faulty variables. The PRBMCA framework is a generic formulation in that it is applicable to various MSPM models. The effectiveness and computational efficiency of the proposed methodology is demonstrated through a numerical example and a benchmark problem of the Tennessee Eastman process.

Keywords-Fault detection and isolation; L\textsubscript{1} – penalty; Multivariate contribution analysis; Multivariate statistical process monitoring; Reconstruction-based contribution

1 Introduction

Multivariate statistical process monitoring (MSPM) is a well-known technique for fault detection and diagnosis (FDD) in industrial processes\textsuperscript{[1-7]}. With the development of sensor technology, distributed process control and data acquisition system, more and more process variables can be measured and process operational data can be acquired and stored. MSPM aims to use these data to monitor process operation and detect abnormal situations\textsuperscript{[1,6,8]}. Principal component analysis (PCA) and partial least squares (PLS) are the most commonly used methods in MSPM for reducing the dimension of the process data under normal operation condition (NOC), establishing the monitoring model and control limits\textsuperscript{[8,9]}. Probabilistic PCA
(PPCA) is a probabilistic framework of PCA which provides a low dimension probabilistic density space and a unified monitoring statistic\[10,11\]. In comparison with multiple monitoring statistics schemes, e.g. conventional PCA, the unified monitoring statistic scheme reduces the number of charts to be observed and decreases the false alarm level\[12,13\]. Thus, PPCA provides clearer monitoring results. With the extensions of PCA/PPCA/PLS, e.g. kernel multi-block PLS\[14\], kernel PPCA\[15\], independent component analysis\[16\], PPCA mixture model\[12\], among others, MSPM could be used for processes with complicated attributes such as large scale, non-Gaussian, nonlinear and etc. MSPM have been successfully used in many industrial applications\[17-19\].

Fault isolation is the ‘downstream’ step of fault detection in MSPM\[6,8,20\]. Contribution analysis is the most popular fault isolation method in MSPM\[21,22\]. The conventional contribution analysis, namely complete decomposition contributions\[23\], decomposes the monitoring statistic as the sum of the variable contributions. The variable having the largest contribution to the monitoring statistic that violated the corresponding control limit is considered to have the strongest relationship with the fault. However, this method has two severe shortcomings. It suffers from the “fault smearing” in some cases\[24\], furthermore, it is designed for PCA model and is difficult to be extended for more advanced statistical models developed recently. To address this issue, alternative methods have been proposed, such as reconstruction-based contribution (RBC)\[24\] and missing variable contribution (MVC)\[12\]. In RBC, the amount of reconstruction of the monitoring statistics along a variable direction is the variable’s contribution. Similarly, in MVC, each variable is treated as if it was missing and the reduction of expected (with respect to missing variable) monitoring statistic is the contribution. The variables corresponding to the largest contribution is considered faulty. Moreover, both of these methods are generic in that they are applicable to various statistical models\[12,25,26\].

The underlining process faults usually have complex impact on more than one process variables. The univariate contribution analysis is not suitable for analysing multivariate process faults. In some studies, historical faulty data are used to define possible faulty scenarios by using, e.g. fault angles\[27\], fault subspaces\[28\], fuzzy IF-THEN rules\[29\], so that the future faults can be isolated. However, these faulty data, served as the basis for these methods, are usually difficult to acquire by either real operations or simulations. Meanwhile, all those methods may face difficulties in handling unknown faults.

Multivariate contribution analysis isolates the faults through identifying the most responsible faulty process variable set without any prior knowledge of faults. Although this set of faulty variables may not directly reveal the root cause, it is undoubtedly helpful in pinpointing out the inconsistent variables that should be further investigated. A branch and bound (BAB) algorithm was proposed to automatically isolate faulty variables for the case of a single PPCA\[30\]. Later, a more generic framework named reconstruction based multivariate contribution analysis (RBMCAl) and the corresponding solution strategy by using the BAB method is developed\[31\]. In the RBMCAl framework, a feasible solution set S was defined and the variable combination in S with the largest amount of reconstruction of monitoring statistic was considered as the faulty variable set. Then the combinatorial optimisation problem was solved by BAB algorithm. This framework only concerns with the reconstructed monitoring statistics, but does not with the ‘cost’ on reconstruction. This may results in incorrect fault isolation, especially when the fault magnitude is small or moderate. Meanwhile, the efficiency of this strategy is too low to be implemented in on-line process monitoring.
This paper proposes a penalized reconstruction-based multivariate contribution analysis (PRBMCA) framework to overcome these shortcomings and to improve the computational efficiency of fault isolation. In contrast with RBMCA, the strategy of finding the combination of faulty variables is a two-step scheme combing \( L_1 \) – penalized reconstruction and BAB. In the first step, the \( L_1 \) – penalized optimisation excludes non-faulty variables in an efficient way and obtains a compact set of possible faulty variables. Then, the BAB method of original RBMCA is carried out on the compact set of variables. This two-step scheme has the benefits of high efficiency of \( L_1 \) – penalized optimisation and the advantages of making as fewer variables being reconstructed as possible. Moreover, this is a generic framework without a particular monitoring model. There are various solutions with respect to a wide range of objective functions in the community of \( L_1 \) – penalized optimisation, such that this penalized framework is applicable to various advanced monitoring statistical models \(^{32}\). The proposed PRBMCA framework with the PPCA model is applied to the fault detection and isolation of an illustrative numerical example and the Tennessee Eastman (TE) benchmark problem.

The rest of this paper is organized as follows. In Section 2, fault isolation through RBMCA is reviewed; a sufficient condition for isolatability of faulty variable when using RBMCA is presented as well. The generic penalized reconstruction-based multivariate contribution analysis (PRBMCA) and the application on PPCA model are introduced in Section 3. Section 4 presents the case studies and finally Section 5 concludes this paper.

2 Fault isolation through reconstruction-based multivariate contribution analysis

2.1 Problem formulation

The process data under NOC are always affected by the inevitable process variations, such as measurement noises and fluctuations of external parameters. In the context of MSPM, a statistical model is usually developed to represent the normal process variations. Meanwhile, appropriate monitoring statistic and corresponding control limit are established to distinguish the normal process variations from abnormal disturbances and faults. Without losing generality, we use \( D(x|M) \) to denote the monitoring statistic of \( n \)-dimensional data \( x \) under statistical model \( M \), and CL to denote the corresponding control limit. Generally, the process is considered out-of-control when \( D(x|M) \geq CL \) whereas it is classified as in-control when \( D(x|M) < CL \).

Fault isolation is the down-stream of fault detection in MSPM. If the process sample \( x \) is identified as a faulty sample, i.e. \( D(x|M) \geq CL \), it can be divided into two parts\(^{[8]}\): \( x = x^* + x_f \), the fault free vector is \( x^* \), while the fault is \( x_f = \Xi_{x_f} \cdot f_{x_f} \). Here \( f_{x_f} \) is the fault magnitude and \( \Xi_{x_f} \) is a 0-1 matrix denotes which variables are influenced significantly by the detected fault. For example, if the set of faulty variables is \( X_f = \{x_2\} \) we have...
\[ \Xi_{X_t} = [0 \ 1 \ 0 \ \cdots]^t \]

While if \( X_t = \{x_2, x_4\} \), then
\[ \Xi_{X_t} = [0 \ 1 \ 0 \ 0 \ \cdots]^t \]

Following the concept of fault isolation through reconstruction \(^{24-26}\) (or similarly missing variables approach \(^{12}\)), the reconstruction-based multivariate contribution (RBMVC) corresponding to a given variable set \( X_r \) is
\[
\text{RBMVC}_{X_r} = D(x|\mathbf{M}) - D(x - \Xi_{X_r} \cdot \hat{f}_{X_r} | \mathbf{M})
\]
where \( \hat{f}_{X_r} = \arg\min_{F_{X_r}} D(x - \Xi_{X_r} \cdot f_{X_r} | \mathbf{M}) \). This minimization problem can usually be solved by standard programming approaches \(^{31}\).

It should be noted that the faulty variable set \( X_t \) is unknown for process faults. In fact, the major challenge is to find out appropriate variable set \( X_t \), among \( 2^n \) candidates, to be reconstructed. Ignore the constant \( D(x|\mathbf{M}) \) of the Eq.(1), the RBMVC\(_{X_r}\) depends on the reconstructed monitoring statistic, \( \Phi(X_r) = D(x - \Xi_{X_r} \cdot \hat{f}_{X_r} | \mathbf{M}) \). The objective of fault isolation is to find an appropriate variable set \( \hat{X}_t \), such that the reconstructed monitoring statistic goes back to normal. More rigorously,
\[
\Phi(\hat{X}_t) = D(x - \hat{x}_t | \mathbf{M}) < CL
\]
where \( \hat{x}_t = \Xi_{\hat{X}_t} \cdot \hat{f}_{\hat{X}_t} \).

Finally, RBMCA found out the optimal solution \( S_{RBMCA}^{\text{S}} \) such that:
\[
\Phi(S_{RBMCA}^{\text{S}}) = \min_{X_r \in S} \Phi(X_r)
\]

As there are usually \( 2^n \) candidate sets to be evaluated, a BAB algorithm was proposed in previous work \(^{31}\) to solve this problem efficiently. A solution tree for a problem of five variables is illustrated in Fig.1. Each node of this tree represents a candidate set. The algorithm started from the root node which corresponded to all variables being reconstructed, then gradually made some reconstructed variables fixed according to a BAB strategy. The bounding and pruning operations can significantly save computation time and guarantee the final
solution being the global optimum\cite{31}.

[Fig. 1. Solution tree for five process variables of BAB algorithm.]

2.2 A sufficient condition for isolatability of faulty variables when using RBMCA

The fault isolation through RBMCA depends on the reconstructed monitoring statistic, i.e. $\Phi(X_r)$. Although the simulations in previous study\cite{30} demonstrated the efficiency and effectiveness of the RBMCA framework, there was no any isolatability analysis provided.

**Lemma 1**: For $x_j \in X_n$, if $\Phi\left(X_n \setminus x_j\right) \geq \text{CL}$, then $x_j$ will be isolated as faulty variable by RBMCA.

**Proof**: $\forall x_j \in X_n$, if $x_j \notin X_r$, then $X_r \subseteq \{x_n \setminus x_j\}$. As previous work studied, adding more variables as being reconstructed will always result in no increasing reconstructed value, thus: $\Phi(X_r) \geq \Phi\left(X_n \setminus x_j\right) \geq \text{CL}$. Therefore, $X_r$ does not belong to $S$, nor be isolated by RBMCA.

Then, suppose the faulty variable set is $X_f$, a sufficient condition for isolating $X_f$ correctly by RBMCA, i.e. $X_f = X^\text{RBMCA}_S$, is:

$$\forall x_j \in X_f, \Phi\left(X_n \setminus x_j\right) \geq \text{CL}$$

(5)

**Proof**: According to Lemma 1, $\forall x_j \in X_f$, $x_j$ will be isolated as faulty variables, so $X^\text{RBMCA}_S \supseteq X_f$.

In addition,

$$\Phi(X_f) = D\left(x^* + \varepsilon_{x_i} \cdot f_{x_i} - \xi_{x_i} \cdot \hat{f}_{x_i} | M\right) \leq D\left(x^* + \varepsilon_{x_i} \cdot \left(f_{x_i} - \hat{f}_{x_i}\right) | M\right) = D(x^* | M) < \text{CL}$$

(6)

Thus, if $X^\text{RBMCA}_S \neq X_f$, it is easy to find that $X^\text{RBMCA}_S \notin S(X_f)$ is a subset of $X^\text{RBMCA}_S$, and it can bring the process back to normal). Conclusively, the faulty variables set is $X^\text{RBMCA}_S = X_f$.

In this case, the feasible solution $S$ contains only one variable set $X_f$, because $\forall X_r \in X_n$, $X_r \notin S$, if $\exists x_j \in X_r, x_j \notin X_r$. It could be proved in a similar way that used in the proof of Lemma 1.

In other cases, the faulty variables do not satisfy the sufficient condition; therefore, beside the real solution, there may be other candidates, which namely pseudo solutions in this work, in $S$. For simplicity, suppose $\{x_j\} = X_f$, s.t. $\Phi\left(X_n \setminus x_j\right) < \text{CL}$. In the worst case, $\{X_n \setminus x_j\} \in S$, and the RBMCA would obtain incorrect solution, if $\Phi\left(X_n \setminus x_j\right) < \Phi(X_f)$. In addition, $\Phi\left(X_n \setminus x_j\right)$ usually very small, since it reconstructed much more pseudo faulty variables than those reconstructed regard to $X_f$. As the simulation in section 4.1 shows, obtaining pseudo solutions by RBMCA are frequently, especially when the fault magnitude is moderate or small.

In summary, the RBMCA guarantees that the faulty variable $x_j$ will be isolated, if $\Phi(X_n \setminus x_j) \geq \text{CL}$; otherwise, it may isolate incorrect variables. Generally, $\Phi(X_n \setminus x_j)$ depends on the fault magnitude corresponds to $x_j$, but the analytical expressions of Eq.(5) may not exist for some complicated statistical model $M$. 

\[5\]
3 PRBMCA combining $L_1$-penalized reconstruction and BAB

3.1 Generic PRBMCA framework

As discussed in previous section, obtaining pseudo solutions is a severe shortcoming of RBMCA, especially when the fault magnitude is small or moderate. Moreover, the pseudo solutions are unacceptable in practice, since lots of non-faulty variables were isolated as faulty.

In contrast with real faulty variable set, the pseudo solutions usually reconstruct much more variables; the estimation of fault magnitude is much greater as well. Consequently, that penalizing the reconstructed monitoring statistic by adding constraint on estimation of fault magnitude is an effective way to prevent obtaining the pseudo solutions. In other words, adding a penalized term on $\hat{x}_f$ in the objective function in Eq.(2) while finding the feasible faulty variable sets.

This concept is similar with setting some coefficients to zero in penalized regression. In the community of statistics, the ‘Least Absolute Shrinkage and Selection Operators’ (LASSO) is a popular algorithm to solve this problem\textsuperscript{[33,34]}. LASSO minimizes the sum of squared errors subject to a constraint on the $L_1$ norm of estimators’ vector. This constraint shrinks some of the estimators toward 0, so generates a sparse model that is more easily interpreted. It is essentially a $L_1 -$ penalized optimisation formulation which is used widely in the areas of machine learning \textsuperscript{[32]}, compressed sensing \textsuperscript{[35]} and etc.

Motivated by these works, a penalized RBMCA framework consists two sequential steps is developed in this section. The $L_1 -$penalized reconstruction is carried out to obtain a compact candidate set of fault variables in the first step; then, the BAB algorithm derived in original RBMCA is implemented to isolate the most intrinsic faulty variables in the compact candidate set.

Firstly, the fault isolation is formulated into a $L_1 -$ penalized optimisation problem:

$$\min_{\hat{x}_f} D(x - \hat{x}_f | M)$$

Subject to: $|\hat{x}_f|_{L_1} = \sum_{i=1}^{n} |\hat{x}_{f,i}| < t$ \hspace{1cm} (7)

As discussed in the community of $L_1 -$ penalized optimisation, $t$ is a very crucial tuning parameter. In this paper, it is determined by using a binary search algorithm. The reconstruction should bring the process back to normal with as fewer variables as possible. Furthermore, smaller $t$ usually results in fewer nonzero operators (faulty variables in the context of fault isolation). Thus, the problem can be formulated as:

$$t_{opt} = \inf\{t | D(x - x_f(t) | M) < CL\}$$ \hspace{1cm} (8)

Where $x_f(t) = \arg \min_{\hat{x}_f} D(x - \hat{x}_f | M)$, s.t. $|\hat{x}_f|_{L_1} = \sum_{i=1}^{n} |\hat{x}_{f,i}| < t$.

In addition, $D(x - x_f(t_1) | M) \geq D(x - x_f(t_2) | M)$ , if $t_1 < t_2$. Let $t_{max} = |x_f(+\infty)|_{L_1}$, then $D(x - x_f(t_{max}) | M) = D(x - x_f(+\infty) | M) < CL$. Thus $t_{opt} \in [0, t_{max}]$, and a standard binary search algorithm can be applied to find $t_{opt}$ efficiently.
Algorithm 1:

Initialisation
{
    $t_{\text{lower}} = 0$;
    $t_{\text{upper}} = t_{\text{max}}$;
}

$t$ binary_search ($t_{\text{lower}}$, $t_{\text{upper}}$, threshold)
{
    if ($ t_{\text{upper}} - t_{\text{lower}} <$ threshold
        return $t_{\text{upper}}$;
    else
        {
            $t_{\text{mid}} = (t_{\text{upper}} - t_{\text{lower}})/2$;
            if ($D(\vec{x} - \vec{x}_f(t_{\text{mid}})|\mathbf{M}) \geq \text{CL}$
                return binary_search ($t_{\text{mid}}$, $t_{\text{upper}}$, threshold);
            else if ($D(\vec{x} - \vec{x}_f(t_{\text{mid}})|\mathbf{M}) < \text{CL}$
                return binary_search ($t_{\text{lower}}$, $t_{\text{mid}}$, threshold);
        }
}

A typical value of threshold is $\frac{t_{\text{max}}}{1000}$. This binary search algorithm gets $t_{\text{opt}}$ and $\vec{x}_f(t_{\text{opt}})$ simultaneously.

Denote the set of nonzero indices of $\vec{x}_f(t_{\text{opt}})$ as $L_{1}$.

However, $X_{s}^{L_{1}}$ does not always belong to feasible solution set $S$. In this case, the reconstructed monitoring statistic may be still less than CL after excluding some variables from $X_{s}^{L_{1}}$. In practice, this is unacceptable for engineers since some non-faulty variables are isolated incorrectly. To avoid isolating non-faulty variables, the BAB algorithm proposed in the RBMCA framework is implemented to refine $X_{s}^{L_{1}}$. This two-step scheme guarantees that the final result belongs to $S$; hence, as fewer variables being reconstructed as possible and no non-faulty variable is isolated.

The PRBMCA framework implements through as follows.

Step 1: Using Algorithm 1 to get $X_{s}^{L_{1}}$;

Step 2: The BAB algorithm proposed in the RBMCA framework is implemented to refine the faulty variables. Let $X_{s}^{L_{1}}$ to be the root node in the search tree, corresponding to all the variables of $X_{s}^{L_{1}}$ being reconstructed. Then some of the reconstructed variables are gradually made to be fixed according to the search path determined by the BAB algorithm. Finally, we get the optimal solution $X_{s}^{\text{PRBMCA}}$ such that:

$$\Phi(X_{s}^{\text{PRBMCA}}) = \min_{X_{s} \in S} \Phi(X_{s}), \text{ subject to } X_{s}^{\text{PRBMCA}} \subseteq X_{s}^{L_{1}}$$  \hspace{1cm} (9)

Beside accuracy, efficiency is another important issue in fault isolation. It is crucial that it takes reasonable time to isolate faulty variables, especially in on-line process monitoring. This two-step scheme
improves the efficiency and reduces the computation time cost significantly in contrast with original RBMCA framework. Firstly, in most cases, the $L_1$ penalized optimization problem in Eq.(7) is solved by algorithms with time complexity of $O(n)$, which is much better than $O(2^n)$ of BAB used in RBMCA. Secondly, it excludes lots of pseudo solutions, and yields a compact candidate variables set on which in step 2 the BAB implements; therefore, the time cost of researching for optimal solution by BAB is reduced drastically.

This PRBMCA framework combines $L_1$ penalized reconstruction and BAB to isolate the faulty variables. By narrowing down the candidate faulty variables in the first step through $L_1$ penalizing, it improves the efficiency of solving the combinatorial optimization problem in Eq.(4), so the time cost of fault isolation is reduced drastically than of original RBMCA framework. Meanwhile, it improves the accuracy and effectiveness of fault isolation as well since pseudo solutions, which might be obtained by original RBMCA and are unacceptable in practice, are excluded. In contrast with $L_1$ penalized reconstruction, i.e. implements step 1 only, the BAB of step 2 guarantees that the final solution belongs to feasible solution set S. The case studies in section 4 demonstrate that the PRBMCA framework performs better than the original RBMCA framework and the $L_1$ penalized reconstruction.

3.2 PRBMCA with PPCA model for process monitoring

In the community of MSPM, PCA is the most commonly used technique to reduce the high dimensions of the process data and to establish the statistical model of NOC. Probabilistic PCA (PPCA) \cite{10,11} is a probabilistic formulation of PCA. PPCA projects the original $n$-dimensional data $\mathbf{x}$ onto the $d$-dimensional ($d<n$) latent variable $\mathbf{t}$: $\mathbf{x} = \mathbf{Wt} + \mathbf{\mu} + \mathbf{e}$, where $\mathbf{\mu}$ is the mean of sampled data and $\mathbf{e}$ is the noise term. Specifically, the noise $\mathbf{e}$ is assumed to be Gaussian with diagonal covariance matrix: $\mathbf{e} \sim \mathcal{N}(\mathbf{0}, \sigma_\mathbf{e}^2 \mathbf{I})$, which implies $\mathbf{x} \mid \mathbf{t} \sim \mathcal{N}(\mathbf{Wt} + \mathbf{\mu}, \sigma_\mathbf{e}^2 \mathbf{I})$. Furthermore, by adopting a Gaussian prior distribution for the scores, $\mathbf{t} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, the marginal distribution of the data is also Gaussian: $\mathbf{x} \sim \mathcal{N}(\mathbf{\mu}, \mathbf{C})$, where the covariance matrix is $\mathbf{C} = \mathbf{WW}^T + \sigma_\mathbf{e}^2 \mathbf{I}$. The maximum likelihood estimations of model parameters are:

$$
\mathbf{W}_{\text{ML}} = \left( \mathbf{A}_d - \sigma_{\text{ML}}^2 \mathbf{I} \right)^2
$$

$$
\sigma_{\text{ML}} = \frac{1}{n-d} \sum_{j=d+1}^{n} \lambda_j
$$

where $\mathbf{U}_d$ is the loading matrix composed of the eigenvectors of the covariance matrix corresponding to the largest $d$ eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_d$ in the diagonal matrix $\mathbf{A}_d$, and $\lambda_{d+1}, \lambda_{d+2}, \ldots, \lambda_n$ are the smallest $n-d$ eigenvalues.

The monitoring statistic $D$ based on multivariate Gaussian distribution and a corresponding confidence bound is adopted \cite{12}. In particular a process data sample $\mathbf{x}$ is considered to be out-of-control when:

$$
D = (\mathbf{x} - \mathbf{\mu})^T \mathbf{C}^{-1}(\mathbf{x} - \mathbf{\mu}) > \chi^2_n(\beta)
$$

where $\chi^2_n(\beta)$ is the $\beta$-fractile of the $\chi^2$ distribution with $n$ degrees of freedom.

There are various latent variable models in the community of MSPM to enhance the performance of PPCA in modeling the process and noises, such as robust PPCA\cite{30} to deal with atypical observations by using Student-$t$ distribution, a heavy tailed generation of Gaussian distribution, to instead the original Gaussian model of noise. Nonetheless, the PRBMCA is a generic framework and it is applicable to various statistical
models in the context of MSPM.

In practice, data are usually scaled to zero mean, thus the monitoring statistic is reduced to $D(\mathbf{x}) = \mathbf{x}^T \mathbf{C}^{-1} \mathbf{x}$. In order to apply the PRBMCA to PPCA model, the following optimisation problem should be solved:

$$\min_{\tilde{\mathbf{x}}_f} (\mathbf{x} - \tilde{\mathbf{x}}_f)^T \mathbf{C}^{-1} (\mathbf{x} - \tilde{\mathbf{x}}_f)$$

Subject to: $|\tilde{\mathbf{x}}_f|_{L_1} = \sum_{i=1}^{n} |\tilde{x}_{fi}| < t \quad (13)$

Because of the non-differentiability of $L_1$-norm constraint, the original optimisation problem in Eq.(13) is challenging. An effective transformation was suggested by Tibshirani [33]. Consider two auxiliary vectors $\mathbf{x}_{f+}$ and $\mathbf{x}_{f-}$, $\mathbf{x}_f = \mathbf{x}_{f+} - \mathbf{x}_{f-} = [\mathbf{I}_n - \mathbf{I}_n] \begin{bmatrix} \mathbf{x}_{f+} \\ \mathbf{x}_{f-} \end{bmatrix}$, then the original problem could be transformed as follows. Ignore the constant terms:

$$\min_{\mathbf{x}_{f+}, \mathbf{x}_{f-}} -2\mathbf{x}_f^T \mathbf{C}^{-1} \begin{bmatrix} \mathbf{x}_{f+} \\ \mathbf{x}_{f-} \end{bmatrix} + \begin{bmatrix} \mathbf{x}_{f+} & \mathbf{x}_{f-} \end{bmatrix} \begin{bmatrix} \mathbf{C}^{-1} & -\mathbf{C}^{-1} \\ -\mathbf{C}^{-1} & \mathbf{C}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{x}_{f+} \\ \mathbf{x}_{f-} \end{bmatrix}$$

Subject to: $||\begin{bmatrix} \mathbf{x}_{f+} \\ \mathbf{x}_{f-} \end{bmatrix}||_{L_1} = \sum_{i=1}^{n} x_{f+,i} + \sum_{j=1}^{n} x_{f-,j} < t \quad \forall i, j, x_{f+,i} \geq 0, x_{f-,j} \geq 0 \quad (14)$

Then, the transformed problem in Eq.(14) can be solved by using standard quadratic programming algorithms such as interior point algorithm, active set algorithm.

The $L_1$-penalized optimisation, formulated in Eq.(14), is implemented as the step 1 of PRBMCA framework to obtain a compact candidate faulty variable set $\tilde{\mathbf{X}}_f^{L_1}$. Then, in step 2, the BAB algorithm proposed in generic RBMCA framework, instantiated with PPCA model, is executed to search for the optimal combination of the faulty variables according to Eq.(9).

4 Case Study

In this section, the effectiveness and efficiency of fault isolation by using PRBMCA framework is demonstrated through a numerical example and the TE benchmark problem. The comparison between PRBMCA, $L_1$-penalized reconstruction and RBMCA is provided, especially for small magnitude faults. All computation is carried out using MATLAB® 2011a on a Laptop with Intel® CoreTM Duo CPU T9400 (2.53GHz).

4.1 An illustrative example

A numerical example with twenty variables (x) and two internal states (y) is studied here to illustrate the fault isolation based on PRBMCA. The numerical system is:

$$\mathbf{x} = \mathbf{N} \mathbf{y} + \mathbf{v} \quad (15)$$

where
The noise term \( \mathbf{v} \) is Gaussian distributed with mean zero and variance 0.1. First, 1000 samples are generated under NOC and a PPCA model is established with two principal components. The control limit is 99\% fractile of the \( \chi^2 \) distribution with degree of freedom \( n=20 \), i.e. \( CL = \chi^2_{20}(0.99) = 37.566 \).

To simulate faults, step changes are introduced to the 4\textsuperscript{th}, 8\textsuperscript{th}, 12\textsuperscript{th}, 16\textsuperscript{th} and 20\textsuperscript{th} variables. Moreover, five different step sizes are considered to study the impact of fault magnitude. The step sizes are \( k (k=2, 3, 4, 5, 6) \) times of the standard deviations of the corresponding variables. For each case, 100 faulty samples are examined and two key criterions are adopted to evaluate the performances, including the average computation time and the fault isolation accuracy. The accuracy is the percentage that the faulty variables are isolated correctly. Table 1 shows the results of fault isolation using RBMCA, \( L_1 \)–penalized reconstruction and PRBMCA.

\[
\mathbf{N} = \begin{bmatrix}
-0.47 & 1.54 & -0.54 & 2.14 & 2.02 & -0.81 & 2.68 & 0.91 & -0.31 & 0.89 & 1.36 & 0.99 & -0.25 & 0.24 & 0.07 \\
0.43 & 0.24 & 0.03 & -1.83 & -0.83 & -0.83 & 1.58 & 0.68 & -0.35 & -2.64 & 0.80 & -1.35 & 1.12 & 0.25 & 0.24 & 0.07 \\
0.43 & 0.24 & 0.03 & 1.83 & 0.83 & 0.4 & 1.58 & 0.68 & 0.35 & 2.64 & 0.80 & 1.35 & 1.12 & 0.99 & 0.47 & 1.24 & 0.07 \\
\end{bmatrix}
\]

\[
\mathbf{Y} = \begin{bmatrix}
\gamma_1 \\
\gamma_2 \\
\end{bmatrix}, \gamma_1 \sim \mathcal{N}(2,1), \gamma_1 \sim \mathcal{N}(5,1)
\]

The fault isolation accuracy of PRBMCA is much better than those of \( L_1 \)–penalized reconstruction and RBMCA, especially when the fault magnitude is small or moderate. The isolated results of \( L_1 \)–penalized reconstruction usually contain too many variables and some of these variables would be excluded by BAB algorithm. On the other hand, the results of RBMCA are contaminated by pseudo solutions. The \( L_1 \)–penalized reconstruction procedure included in PRBMCA excludes those pseudo solutions efficiently.

Furthermore, the computation performance of PRBMCA is much better than RBMCA. That the average computation time of PRBMCA for different fault magnitudes are all less than 0.3 s suggests its on-line fault isolation capability. This “instantaneous feature” is crucial for practical application of on-line MSPM. The process engineers can make a faster response based on these isolating results so that the detrimental effect of the fault is minimized.

The performance of fault isolation is also affected by the total number of process variables (\( n \)). Four cases with different number of process variables (\( n=20, 30, 40, 50 \)) are presented here. The linear system for each case is similar to Eq.(15) and Eq.(16) except that the \( n \times 2 \) transform matrix \( \mathbf{N} \) is generated randomly. Then, the step changes are introduced in five randomly selected process variables to simulate the faults. All the step sizes are \( k=6 \). For each case, 100 faulty samples are examined and the fault isolation results are summarized in Table 2.

In comparison with RBMCA and \( L_1 \)–penalized reconstruction, PRBMCA still performs best in accuracy, especially when the number of variables is very large (\( n>30 \)). Moreover, the reduction in computation time of PRBMCA in contrast to RBMCA is very significant. The PRBMCA get results “immediately” after fault
occurred in all cases, even though \( n = 50 \) and the size of original multivariate contribution analysis problem is \( 2^{50} \approx 1.13 \times 10^{15} \). The average computation time of RBMCA for \( n = 50 \) is nearly one hundred times than PRBMCA.

Moreover, the number of faulty variables \( (n_f) \) is another crucial factor for fault isolation. For \( n = 50 \), different number of faulty variables \( (n_f = 5, 10, 15, 20, 25, 30, 35, 40, 45) \) are examined, all the fault magnitudes are \( k = 6 \). The computation performances are given in Table 3. These results are still averaged over 100 random simulations of each case.

Table 3. Computation performances of fault isolation using PRBMCA with different number of faulty variables \( (k=6,n=50) \)

Even for the worst case of \( n_f = 45 \), the average computation time of PRBMCA is no more than few seconds (the maximal average computation time is 16.268 seconds). Moreover, the performance of PRBMCA for \( n_f < 35 \) is quite sound, the maximum computation time is only about three seconds.

This numerical example shows that the proposed PRBMCA is suitable for fault isolation on static models. However, these static models are not adequate to represent real industrial processes and faults. As process dynamics cause fault propagation, the faulty variable sets are varied with time. This sequence of faulty variable sets provides crucial information for engineers to identify the root cause of the detected fault. In next section, fault isolation by using PRBMCA for the TE benchmark problem is presented to demonstrate the effectiveness and efficiency of PRBMCA to capture the propagations.

4.2 Tennessee Eastman benchmark problem

The TE process is a well-known benchmark problem for testing process control strategies and MSPM methods [37]. The flow sheet of the TE process is shown in Fig. 2. This plant has five units: reactor, condenser, separator, stripper and compressor. This process produces two products (G and H) from four reactants (A, C, D and E). A byproduct F is also produced. In addition, an inert component B presents in C stream (stream 4) and in trace amount in A feed stream (stream 1). The reactions are:

\[
\begin{align*}
A(g) + C(g) + D(g) & \rightarrow G(\text{liq}), \text{Product 1} \\
A(g) + C(g) + E(g) & \rightarrow H(\text{liq}), \text{Product 2} \\
A(g) + E(g) & \rightarrow F(\text{liq}), \text{Byproduct} \\
3D(g) & \rightarrow 2F(\text{liq}), \text{Byproduct}
\end{align*}
\]

The original process is open-loop and unstable. The reactor pressure grows quickly and exceeds the safe upper-limit such that all the plant would be shut down. The decentralized control strategy developed by Ricker [38] is used in this study to regulate the steady-state operation. There are six operation modes according to different mass ratios of G/H. The basic mode 1 is adopted in this simulation.

Fig. 2. Flow sheet of the TE process.
For simplicity, only 22 continuous measurements, listed in Table 4, are selected for MSPM and fault isolation in this study. Four typical process faults listed in Table 5 are simulated to illustrate the fault isolation by using PRBMCA. The comparison between PRBMCA and RBMCA is provided as well. The isolation results are all validated by analysing the physical models and underlying process operations according to the description of root cause of these faults in the original paper.\[30\]

First, 2000 samples under NOC are generated with sampling interval of 0.01hr. Then, a PPCA model with 12 principal components was established based on normal data, and the control limit is 99%-fractile of the \(\chi^2\) distribution with degree of freedom \(n=22\), i.e. \(CL = \chi^2_{22}(0.99) = 40.289\).

The process runs for two hours and produces 200 faulty samples in each fault scenario. Fault isolation is carried out on a sequence of faulty samples to reveal the fault propagation path in the plant. Although the process is approximately static at normal operation condition, it is unstable and varied with time in all fault scenarios. In particularly, the propagation of the faulty variables reflects the process dynamics in fault scenarios; thus it provides crucial information for fault diagnosis and root cause analysis. The control charts of the four process faults are depicted in Fig. 3 to Fig. 6. The fault isolation results of PRBMCA and RBMCA shown in Table 6-9 include the sequence of faulty variable sets, which reveal the faulty variable propagations, and the computation time.

Table 4. Continuously measured variables

Table 5. Operational faults

As the PRBMCA is designed for on-line fault isolation, the computation should be finished within a reasonable time. In this study, the case which takes more than 5 minutes are considered as unacceptable and marked as “failed”.

Fig. 3. Monitoring results of process fault 4.

For fault 4, the monitoring statistics violate the control limit from the 2nd to the 40th samples in Fig.3. The fault isolation is carried out from 2nd to 9th faulty samples, and the results are shown in Table 6.

Table 6. Fault isolation results of fault 4

The fault variables obtained by both methods are the same, but the average computation time of PRBMCA (0.296s) is much less than that of RBMCA (4.092s). Moreover, the RBMCA failed on the 6th fault sample, i.e. the computation time is longer than five minutes. Meanwhile, all the computation through PRBMCA on eight fault samples are less than one second, thus it is applicable in time-critical situations.

Further inspection of process operation confirms that the results are reasonable. A step change in reactor cooling water inlet temperature directly leads to abnormal variation of reactor temperature \(x_9\) and reactor cooling water outlet temperature \(x_{21}\). Later, cooling water out temperature \(x_{21}\) temporarily goes back to normal, because the control loop turns on the valve and the flow of cooling water is increased. Then the change of reactor temperature leads to change in reaction rate and, as a results, other related process variables including Reactor Pressure (\(x_7\)), Product Separator Pressure (\(x_{13}\)), and Stripper Pressure (\(x_{16}\)) gradually deviate from their NOC values.
For fault 6, the monitoring results are shown in Fig. 4. Fault isolation is carried out from 1st to 8th faulty samples, and the results are shown in Table 7.

[Fig. 4. Monitoring results of process fault 6.]

Obviously, the efficiency of PRBMCA is much higher than that of RBMCA. The average computation time (0.242s) of PRBMCA is much less than that of RBMCA (8.49s). Meanwhile, most of the identified faulty variables by PRBMCA are the same as those by RBMCA except for faulty sample No.3.

[Table 7. Fault isolation results of fault 6]

All these results can be validated by analysing process dynamic properties. The root cause of fault 6 is the loss of A feed. Thus the flow rate of component A in stream 1 (x1) is immediately affected. Then, lack of reactant A cause reaction releases less heat (main reactions are exothermic), and thus the reactor temperature (x9) decrease and the reactor cooling water outlet temperature (x21) is also changed. Notice that the cooling water control mechanism would maintain the temperature of reactor (x9) stay in NOC, thus it is not isolated as faulty variable in some faulty samples. Meanwhile, the imbalance reactants would lead to a rapid pressure increase in the reactor (x7), in the product separator (x13) and in the stripper (x16), and later the purge rate (x10) is increased to reduce the pressure in the plant. The different result between PRBMCA and RBMCA is faulty variables for faulty sample No.3. However, both of these results are reasonable since the temperature and pressure in the plant are both influenced by this fault.

Fig.5 shows the monitoring results of fault 7. Fault isolation is carried out from 1st to 8th faulty samples, and the results are shown in Table 8. The average computation time (0.312s) of PRBMCA is less than that of RBMCA (2.847s). Meanwhile, all of the identified faulty variables by PRBMCA are the same as those by RBMCA.

Further, more (1st to 200th) faulty samples were examined to investigate the propagation of fault variables in a longer period. The sequence of identified variables is summarized as: \{x_4\}→\{x_4, x_7, x_9, x_{10}, x_{13}, x_{16}, x_{18}, x_{21}\}→\{x_7, x_{10}, x_{11}, x_{13}, x_{16}, x_{18}, x_{22}\}→\{x_7, x_{10}, x_{13}, x_{16}\}→\{x_{10}\}. The average computation time of PRBMCA (0.375s) is significantly less than of RBMCA (21.201s).

[Fig. 5. Monitoring results of process fault 7.]

[Table 8. Fault isolation results of fault 7]

The root cause of fault 7 involves C header pressure loss (reduced availability) in stream 4, resulting in a decrease in the total feed in stream 4 (x4). Then, similarly with the scenario of fault 6, this disturbance causes variation in several pressure and temperature variables, such as x7, x9, x_{11}, x_{13}, x_{16} and x_{18}, due to insufficient reaction. Later cooling water temperatures (x_{21}, x_{22}) are affected as well. Further, these variables gradually go back to NOC because of the actions of process control system. Finally, only purge rate (x_{10}) is higher than normal to remove excessive by-products that results from insufficient main reactions.

For fault 11, the monitoring results are shown in Fig. 6. Fault isolation is carried out from 14th to 21st samples, and the results are shown in Table 9. As expected, the efficiency of PRBMCA is much better than of RBMCA (average computation time is 0.279s vs. 41.354s). The results can be validated in the same way as that used in validating results of the scenario of fault 4, because both the root causes of the faults are changes in the inlet temperature of reactor cooling water.

[Fig. 6. Monitoring results of process fault 11.]
As comparison, conventional PCA-based contribution analysis and univariate reconstruction-based contribution analysis is carried out for sample number 4 of fault 7. Firstly, Fig.7 shows the conventional PCA-based contributions. Obviously, it is hardly to determine which variables are the dominant sources of the fault from these contributions; because there is no any contribution is significant. Besides, the univariate reconstruction-based contributions are shown in Fig.8. Notice that the contributions and control limits are all shown in logarithm scale for ease of illustration. More rigorously, univariate contributions are log(RBVMC_{i}), i = 1,2⋯n, the Log-Control limit is log(D(x|M) − CL), where D(x|M) is the monitoring statistic of sample number 4 and CL = χ^2_{22}(0.99) = 40.289. Similarly, no univariate contribution exceeds the control limit, suggesting that univariate reconstruction-based contribution analysis does not unambiguously reveal the faulty variables. It may be argued that the variable 4 with the largest contribution is faulty, but it cannot ignore that lots of variables’ contributions are very close to that of variable 4; therefore, they cannot be excluded from the candidate set of faulty variables. This ambiguity, with regard to which combination of variables should be isolated, is exactly the issue that the PRBMCA and original RBMCA address. Moreover, the PRBMCA proposed in this work isolate the faulty variables much more effective and efficient than original RBMCA.

5 Conclusions

A generic fault isolation framework, namely PRBMCA, combining L_1 – penalized reconstruction and BAB algorithms is proposed in this paper. Following the concept of RBMCA, process variables (and combinations of variables) are reconstructed to minimize the monitoring statistic. In contrast to original RBMCA, this penalized framework consists of two sequential steps. In the first step, L_1 – penalized reconstruction excludes pseudo solutions and obtains a compact set of candidate faulty variables in an efficient way. Then, BAB algorithm refines the compact set and isolates faulty variables. This penalized framework is applied to PPCA model, and its effectiveness and efficiency is demonstrated through a numerical example and Tennessee Eastman benchmark problem, especially in contrast to original RBMCA. The case studies demonstrate that PRBMCA isolates the correct faulty variables much faster than RBMCA without loss in accuracy in all cases. Moreover, the sequences of faulty variables provide crucial information about the dynamic features of process faults. Thus future research could be carried out on developing a formal framework to investigate the fault propagation information for diagnosis and root cause analysis purposes.

Acknowledgement

The work is supported by Tsinghua University (Tsinghua Scholarship for Overseas Graduate Studies) and EU through the project iREMO (grant No. NMP2-SL-2009-228662). The authors are grateful to Professor David SH Wong at National Tsing Hua University for discussions on using LASSO to solve combinatorial optimisation problems.
Abbreviations

BAB    Branch and Bound;
CL     Control Limit;
FDD    Fault Detection and Diagnosis;
LASSO  Least Absolute Shrinkage and Selection Operator;
MSPM   Multivariate Statistical Process Monitoring;
MVC    Missing Variable Contribution;
NOC    Normal Operation Condition;
PCA    Principal Component Analysis;
PPCA   Probabilistic Principal Component Analysis;
PRBMCA Penalized Reconstruction-based Multivariate Contribution Analysis;
RBC    Reconstruction-based Contribution;
RBMCA  Reconstruction-based Multivariate Contribution Analysis;
SPE    Squared Predication Error;

\( \mathbf{x}^* \)    The fault free part of sample \( \mathbf{x} \);
\( \mathbf{x}_f \)    The faulty part of sample \( \mathbf{x} \);
\( \hat{\mathbf{x}}_f \) The estimation of faulty part of sample \( \mathbf{x} \);
\( \mathbf{x}_n \)    The ensemble of all process variables;
\( \mathbf{x}_f \)    Faulty variable set;
\( \mathbf{x}_r \)    Reconstructed variable set;
\( \mathbf{z}_{\mathbf{x}_f} \) The 0-1 matrix corresponding to \( \mathbf{x}_f \);
\( \mathbf{f}_{\mathbf{x}_f} \) The fault magnitude vector;
\( \hat{\mathbf{f}}_{\mathbf{x}_f} \) The estimation of fault magnitude by RBMVC corresponding to \( \mathbf{x}_r \);
\( \Phi(\mathbf{X}_r) \) Reconstructed monitoring statistic corresponding to \( \mathbf{x}_r \);
\( \mathbf{x}_f(\mathbf{t}) \) The estimation of faulty part by \( \mathbf{L}_1 \)-penalized reconstruction corresponding to tuning parameter \( t \);
\( \mathbf{x}_{f+} \) Auxiliary vector used in \( \mathbf{L}_1 \)-penalized reconstruction;
\( \mathbf{x}_{f-} \) Auxiliary vector used in \( \mathbf{L}_1 \)-penalized reconstruction;
\( t_{opt} \) The optimal of tuning parameter \( t \) in \( \mathbf{L}_1 \)-penalized reconstruction;
$X^\text{RBMCA}_S$ Solution of RBMCA;

$X^\text{L1}_S$ Solution of $L_1$—penalized reconstruction;

$X^\text{PRBMCA}_S$ Solution of PRBMCA;

References


[27] Raich, A.; Cinar, A. Diagnosis of process disturbances by statistical distance and angle measure, Computers and Chemical Engineering. 1997, 6, 661-673.


[33] Tibshirani, R. Regression shrinkage and selection via the Lasso, Journal of the Royal Statistical Society B.


List of Figures

Fig. 1. Solution tree for five process variables of BAB algorithm.
Fig. 2. Flow sheet of the TE process.
Fig. 3. Monitoring results of process fault 4.

Fig. 4. Monitoring results of process fault 6.
Fig. 5. Monitoring results of process fault 7.

Fig. 6. Monitoring results of process fault 11.
Fig. 7. SPE and $T^2$ contribution plots of sample number 4, Fault 7.

Fig. 8. Univariate reconstruction-based contributions of sample number 4, Fault 7.
### List of Tables

Table 1. Results of fault isolation using RBMCA, $L_1$-penalized reconstruction, and PRBMCA ($n=20$)

<table>
<thead>
<tr>
<th>Case</th>
<th>RBMCA</th>
<th>$L_1$-penalized reconstruction</th>
<th>PRBMCA</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k=2$</td>
<td>Accuracy</td>
<td>24%</td>
<td>14%</td>
</tr>
<tr>
<td></td>
<td>Computation time</td>
<td>32.647 s</td>
<td>0.192 s</td>
</tr>
<tr>
<td>$k=3$</td>
<td>Accuracy</td>
<td>37%</td>
<td>28%</td>
</tr>
<tr>
<td></td>
<td>Computation time</td>
<td>10.793 s</td>
<td>0.208 s</td>
</tr>
<tr>
<td>$k=4$</td>
<td>Accuracy</td>
<td>51%</td>
<td>45%</td>
</tr>
<tr>
<td></td>
<td>Computation time</td>
<td>4.288 s</td>
<td>0.182 s</td>
</tr>
<tr>
<td>$k=5$</td>
<td>Accuracy</td>
<td>62%</td>
<td>42%</td>
</tr>
<tr>
<td></td>
<td>Computation time</td>
<td>1.158 s</td>
<td>0.239 s</td>
</tr>
<tr>
<td>$k=6$</td>
<td>Accuracy</td>
<td>97%</td>
<td>61%</td>
</tr>
<tr>
<td></td>
<td>Computation time</td>
<td>0.337 s</td>
<td>0.245 s</td>
</tr>
</tbody>
</table>

Table 2. Results of fault isolation using RBMCA, $L_1$-penalized reconstruction, and PRBMCA ($k=6$)

<table>
<thead>
<tr>
<th>Case</th>
<th>RBMCA</th>
<th>$L_1$-penalized reconstruction</th>
<th>PRBMCA</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n=20$</td>
<td>Accuracy</td>
<td>97%</td>
<td>61%</td>
</tr>
<tr>
<td></td>
<td>Computation time</td>
<td>0.337 s</td>
<td>0.245 s</td>
</tr>
<tr>
<td>$n=30$</td>
<td>Accuracy</td>
<td>92%</td>
<td>42%</td>
</tr>
<tr>
<td></td>
<td>Computation time</td>
<td>1.167 s</td>
<td>0.378 s</td>
</tr>
<tr>
<td>$n=40$</td>
<td>Accuracy</td>
<td>85%</td>
<td>45%</td>
</tr>
<tr>
<td></td>
<td>Computation time</td>
<td>12.407 s</td>
<td>0.456 s</td>
</tr>
<tr>
<td>$n=50$</td>
<td>Accuracy</td>
<td>82%</td>
<td>36%</td>
</tr>
<tr>
<td></td>
<td>Computation time</td>
<td>85.389 s</td>
<td>0.881 s</td>
</tr>
</tbody>
</table>
Table 3. Computation performances of fault isolation using PRBMCA with different number of faulty variables

\[(k=6, n=50)\]

<table>
<thead>
<tr>
<th>Case</th>
<th>(n_f=5)</th>
<th>(n_f=10)</th>
<th>(n_f=15)</th>
<th>(n_f=20)</th>
<th>(n_f=25)</th>
<th>(n_f=30)</th>
<th>(n_f=35)</th>
<th>(n_f=40)</th>
<th>(n_f=45)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time(Min)</td>
<td>0.577s</td>
<td>0.577s</td>
<td>0.530s</td>
<td>0.625s</td>
<td>0.608s</td>
<td>0.577s</td>
<td>0.624s</td>
<td>0.624s</td>
<td>0.624s</td>
</tr>
<tr>
<td>Time(Average)</td>
<td>0.922s</td>
<td>0.928s</td>
<td>0.831s</td>
<td>0.740s</td>
<td>0.996s</td>
<td>0.916s</td>
<td>1.636s</td>
<td>2.204s</td>
<td>16.268s</td>
</tr>
<tr>
<td>Time(Max)</td>
<td>1.061s</td>
<td>1.154s</td>
<td>1.155s</td>
<td>1.155s</td>
<td>2.964s</td>
<td>2.995s</td>
<td>64.462s</td>
<td>120.411s</td>
<td>310.847s</td>
</tr>
</tbody>
</table>

Table 4. Continuously measured variables

<table>
<thead>
<tr>
<th>ID</th>
<th>Description</th>
<th>ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(x_1)</td>
<td>A Feed(Stream1)</td>
<td>(x_{12})</td>
<td>Product Separator Level</td>
</tr>
<tr>
<td>(x_2)</td>
<td>D Feed(Stream2)</td>
<td>(x_{13})</td>
<td>Product Separator Pressure</td>
</tr>
<tr>
<td>(x_3)</td>
<td>E Feed(Stream3)</td>
<td>(x_{14})</td>
<td>Product Separator Underflow(Stream 10)</td>
</tr>
<tr>
<td>(x_4)</td>
<td>A and C Feed(Stream4)</td>
<td>(x_{15})</td>
<td>Stripper Level</td>
</tr>
<tr>
<td>(x_5)</td>
<td>Recycle Flow(Stream 8)</td>
<td>(x_{16})</td>
<td>Stripper Pressure</td>
</tr>
<tr>
<td>(x_6)</td>
<td>Reactor Feed Rate(Stream6)</td>
<td>(x_{17})</td>
<td>Stripper Underflow(Stream 11)</td>
</tr>
<tr>
<td>(x_7)</td>
<td>Reactor Pressure</td>
<td>(x_{18})</td>
<td>Stripper Temperature</td>
</tr>
<tr>
<td>(x_8)</td>
<td>Reactor Level</td>
<td>(x_{19})</td>
<td>Stripper Stream Flow</td>
</tr>
<tr>
<td>(x_9)</td>
<td>Reactor Temperature</td>
<td>(x_{20})</td>
<td>Compressor Work</td>
</tr>
<tr>
<td>(x_{10})</td>
<td>Purge Rate(Stream9)</td>
<td>(x_{21})</td>
<td>Reactor Cooling Water Outlet Temperature</td>
</tr>
<tr>
<td>(x_{11})</td>
<td>Product Separator Temperature</td>
<td>(x_{22})</td>
<td>Condenser Cooling Water Outlet Temperature</td>
</tr>
</tbody>
</table>

Table 5. Operational faults

<table>
<thead>
<tr>
<th>Fault ID</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Reactor Cooling Water Inlet Temperature</td>
<td>Step</td>
</tr>
<tr>
<td>6</td>
<td>A Feed Loss (stream 1)</td>
<td>Step</td>
</tr>
<tr>
<td>7</td>
<td>C Header Pressure Loss-reduced availability (stream 4)</td>
<td>Step</td>
</tr>
<tr>
<td>11</td>
<td>Reactor Cooling Water Inlet Temperature</td>
<td>Random Variation</td>
</tr>
</tbody>
</table>

Table 6. Fault isolation results of fault 4

<table>
<thead>
<tr>
<th>Faulty sample number</th>
<th>Fault variables</th>
<th>PRBMCA</th>
<th>Computation time</th>
<th>Fault variables</th>
<th>RBMCA</th>
<th>Computation time</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>{x_9, x_{21}}</td>
<td>0.187s</td>
<td>{x_9, x_{21}}</td>
<td>0.031s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>{x_9, x_{21}}</td>
<td>0.124s</td>
<td>{x_9, x_{21}}</td>
<td>0.010s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>{x_9, x_{21}}</td>
<td>0.140s</td>
<td>{x_9, x_{21}}</td>
<td>13.463s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>{x_9}</td>
<td>0.171s</td>
<td>{x_9}</td>
<td>0.010s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>{x_9, x_{11}, x_{20}}</td>
<td>0.280s</td>
<td>failed</td>
<td>&gt;300s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>{x_9, x_{21}}</td>
<td>0.764s</td>
<td>{x_9, x_{21}}</td>
<td>0.031s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>{x_9, x_{11}, x_{21}}</td>
<td>0.234s</td>
<td>{x_9, x_{11}, x_{21}}</td>
<td>12.433s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>{x_7, x_9, x_{11}, x_{16}, x_{21}}</td>
<td>0.468s</td>
<td>{x_7, x_9, x_{11}, x_{16}, x_{21}}</td>
<td>2.667s</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 7. Fault isolation results of fault 6

<table>
<thead>
<tr>
<th>Faulty sample number</th>
<th>PRBMCA</th>
<th>Time</th>
<th>RBMCA</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>{x_1}</td>
<td>0.140s</td>
<td>{x_1}</td>
<td>0.047s</td>
</tr>
<tr>
<td>2</td>
<td>{x_1}</td>
<td>0.156s</td>
<td>{x_1}</td>
<td>0.031s</td>
</tr>
<tr>
<td>3</td>
<td>{x_1, x_9, x_{21}}</td>
<td>0.358s</td>
<td>{x_1, x_7, x_{13}, x_{16}}</td>
<td>14.071s</td>
</tr>
<tr>
<td>4</td>
<td>{x_1, x_7, x_{10}, x_{13}, x_{16}}</td>
<td>0.484s</td>
<td>{x_1, x_7, x_{10}, x_{13}, x_{16}}</td>
<td>20.889s</td>
</tr>
<tr>
<td>5</td>
<td>{x_1, x_7, x_{13}, x_{16}, x_{21}}</td>
<td>0.156s</td>
<td>{x_1, x_7, x_{13}, x_{16}, x_{21}}</td>
<td>10.748s</td>
</tr>
<tr>
<td>6</td>
<td>{x_1, x_7, x_9, x_{13}, x_{16}, x_{21}}</td>
<td>0.281s</td>
<td>{x_1, x_7, x_9, x_{13}, x_{16}, x_{21}}</td>
<td>1.872s</td>
</tr>
<tr>
<td>7</td>
<td>{x_1, x_7, x_{13}, x_{16}, x_{21}}</td>
<td>0.140s</td>
<td>{x_1, x_7, x_{13}, x_{16}, x_{21}}</td>
<td>0.031s</td>
</tr>
<tr>
<td>8</td>
<td>{x_1, x_7, x_{10}, x_{13}, x_{16}, x_{21}}</td>
<td>0.218s</td>
<td>{x_1, x_7, x_{10}, x_{13}, x_{16}, x_{21}}</td>
<td>20.233s</td>
</tr>
</tbody>
</table>

Table 8. Fault isolation results of fault 7

<table>
<thead>
<tr>
<th>Faulty sample number</th>
<th>PRBMCA</th>
<th>Time</th>
<th>RBMCA</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>{x_4}</td>
<td>0.328s</td>
<td>{x_4}</td>
<td>0.001s</td>
</tr>
<tr>
<td>2</td>
<td>{x_4, x_7, x_9, x_{10}, x_{13}, x_{16}, x_{18}, x_{20}}</td>
<td>0.250s</td>
<td>{x_4, x_7, x_9, x_{10}, x_{13}, x_{16}, x_{18}, x_{20}}</td>
<td>11.809s</td>
</tr>
<tr>
<td>3</td>
<td>{x_4, x_7, x_9, x_{10}, x_{13}, x_{16}, x_{18}, x_{21}}</td>
<td>0.374s</td>
<td>{x_4, x_7, x_9, x_{10}, x_{13}, x_{16}, x_{18}, x_{21}}</td>
<td>1.139s</td>
</tr>
<tr>
<td>4</td>
<td>{x_4, x_7, x_9, x_{10}, x_{13}, x_{16}, x_{18}, x_{21}}</td>
<td>0.452s</td>
<td>{x_4, x_7, x_9, x_{10}, x_{13}, x_{16}, x_{18}, x_{21}}</td>
<td>0.671s</td>
</tr>
<tr>
<td>5</td>
<td>{x_4, x_7, x_9, x_{10}, x_{13}, x_{16}, x_{18}, x_{21}}</td>
<td>0.218s</td>
<td>{x_4, x_7, x_9, x_{10}, x_{13}, x_{16}, x_{18}, x_{21}}</td>
<td>0.406s</td>
</tr>
<tr>
<td>6</td>
<td>{x_4, x_7, x_9, x_{10}, x_{13}, x_{16}, x_{18}, x_{21}}</td>
<td>0.187s</td>
<td>{x_4, x_7, x_9, x_{10}, x_{13}, x_{16}, x_{18}, x_{21}}</td>
<td>0.062s</td>
</tr>
<tr>
<td>7</td>
<td>{x_4, x_7, x_9, x_{10}, x_{13}, x_{16}, x_{18}, x_{21}}</td>
<td>0.218s</td>
<td>{x_4, x_7, x_9, x_{10}, x_{13}, x_{16}, x_{18}, x_{21}}</td>
<td>7.971s</td>
</tr>
<tr>
<td>8</td>
<td>{x_4, x_6, x_7, x_9, x_{10}, x_{13}, x_{16}, x_{18}, x_{20}, x_{21}}</td>
<td>0.468s</td>
<td>{x_4, x_6, x_7, x_9, x_{10}, x_{13}, x_{16}, x_{18}, x_{20}, x_{21}}</td>
<td>0.718s</td>
</tr>
</tbody>
</table>

Table 9. Fault isolation results of fault 11

<table>
<thead>
<tr>
<th>Faulty sample number</th>
<th>PRBMCA</th>
<th>Time</th>
<th>RBMCA</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>{x_{21}}</td>
<td>0.874s</td>
<td>{x_{21}}</td>
<td>22.886s</td>
</tr>
<tr>
<td>15</td>
<td>{x_9, x_{21}}</td>
<td>0.281s</td>
<td>{x_9, x_{21}}</td>
<td>10.780s</td>
</tr>
<tr>
<td>16</td>
<td>{x_9, x_{21}}</td>
<td>0.187s</td>
<td>{x_9, x_{21}}</td>
<td>228.822s</td>
</tr>
<tr>
<td>17</td>
<td>{x_9, x_{21}}</td>
<td>0.187s</td>
<td>{x_9, x_{21}}</td>
<td>66.737s</td>
</tr>
<tr>
<td>18</td>
<td>{x_9, x_{21}}</td>
<td>0.156s</td>
<td>{x_9, x_{21}}</td>
<td>1.544s</td>
</tr>
<tr>
<td>19</td>
<td>{x_9, x_{21}}</td>
<td>0.172s</td>
<td>{x_9, x_{21}}</td>
<td>0.031s</td>
</tr>
<tr>
<td>20</td>
<td>{x_9, x_{21}}</td>
<td>0.187s</td>
<td>{x_9, x_{21}}</td>
<td>0.001s</td>
</tr>
<tr>
<td>21</td>
<td>{x_9, x_{21}}</td>
<td>0.187s</td>
<td>{x_9, x_{21}}</td>
<td>0.031s</td>
</tr>
</tbody>
</table>