Journal of Process Control, Volume 20, Issue 10, December 2010, Pages 1198-1206

# A Branch and Bound Method for Isolation of Faulty Variables through Missing Variable Analysis

Vinay Kariwala[†], Pabara-Ebiere Odiowei[‡], Yi Cao[‡]*and Tao Chen[†]

[†] School of Chemical and Biomedical Engineering,
Nanyang Technological University, Singapore 637459, Singapore
[‡]School of Engineering, Cranfield University, Cranfield, Bedford MK43 0AL, UK

This version: June 17, 2010

**Abstract**

Fault detection and diagnosis is a critical approach to ensure safe and efficient operation of manufacturing and chemical processing plants. Although multivariate statistical process monitoring has received considerable attention, investigation into the diagnosis of the source or cause of the detected process fault has been relatively limited. This is partially due to the difficulty in isolating multiple variables, which jointly contribute to the occurrence of fault, through conventional contribution analysis. In this work, a method based on probabilistic principal component analysis is proposed for fault isolation. Furthermore, a branch and bound method is developed to handle the combinatorial nature of problem involving finding the contributing variables, which are most likely to be responsible for the occurrence of fault. The efficiency of the method proposed is shown through benchmark examples, such as Tennessee Eastman process, and randomly generated cases.

*Keywords*: Branch and bound, combinatorial optimization, global optimization, multivariate contribution analysis, multivariate statistical process monitoring, principal component analysis.

*Corresponding Author: Tel: +44-1234-750111; Fax: +44-1234-754685; E-mail:y.cao@cranfield.ac.uk

# 1 Introduction

Fault detection and diagnosis (FDD) is a critical approach to ensure safe and efficient operation of manufacturing and chemical processing plants. The available current FDD techniques can be classified into the following three major categories: quantitative model-based, qualitative model-based and process history based methods [1]. Among these techniques, multivariate statistical process monitoring (MSPM), which is based on historical process data, has received considerable attention in terms of both methodological research and industrial applications [2, 3]. The success of MSPM may be attributed to the fact that a large amount of historical process data is usually available, and thus the developed model attains high accuracy in detecting any deviation from established normal operating conditions (NOC).

In traditional MSPM, the principal component analysis (PCA) or the partial least squares (PLS) are applied to model the data collected under NOC. Subsequently, monitoring statistics like Hotelling's $T^2$ and the squared prediction error (SPE) are used for fault detection. During the past few years, the focus of the research in MSPM has been on developing sophisticated statistical models to obtain more realistic representation of the process behaviour. Some notable progress has been made in dealing with dynamic processes [4–8] and process with non-Gaussian distributed data [8–13]. However, investigation into the "downstream" step of MSPM, that is, the diagnosis of the source or cause of the detected fault, has been relatively limited. The primary tool used for fault diagnosis is contribution analysis, which quantifies the contribution of individual variables to the $T^2$ and SPE [14]. Since both the $T^2$ and SPE are monitored in conventional MSPM, contribution analysis is typically performed for these two statistics. Thus, if the two contribution plots indicate different sets of responsible faulty variables, it can be difficult to resolve these conflicts and the decision is usually subject to the operators' experience. Furthermore, contribution analysis investigates individual variables one by one. Thus, it can be ineffective in isolating multiple variables which jointly contribute to the occurrence of fault and can lead to the identification of incorrect variables as being faulty due to the correlation among variables.

More recently, the need to use a unified monitoring statistic, as opposed to using the $T^2$ and SPE individually, has been well recognized. One way is to combine the $T^2$ and SPE algorithmically [15, 16]; the other approach is to rely on a fully probabilistic model to provide a single likelihood-based statistic [10, 17]. A single monitoring statistic offers a clearer interpretation of the contribution plots. In addition, these studies suggested an alternative approach to quantifying the contribution by using the idea of missing variables [15, 17]. The original idea is that each process variable is treated as if it were missing and the monitoring statistic is re-calculated. This procedure is carried out for all variables and the variable giving the largest reduction of the monitoring statistic is considered to contribute the most to the fault. The missing variable approach opens up the opportunity to analyze the joint effect of a group of variables by treating them as missing. However, the major difficulty is the vast number of possible variable combinations that are required to be evaluated. For example, if 5 out of total 50 variables are suspected to be the contributing source, then the total number of combinations is $2,118,760$. Therefore, Yue and Qin [15] made a restrictive assumption that the combinations of faulty variables are specified *a priori* according to known faulty modes, and thus contribution analysis only needs to be conducted on these combinations.

This paper extends the missing-variable based contribution analysis to consider the joint effect of multiple variables, namely multivariate contribution analysis. In this study, we choose the probabilistic PCA (PPCA) [18, 19] for modelling normal operating data and on-line process monitoring. The probabilistic framework of PPCA provides a single monitoring statistic, which avoids possible conflict in fault isolation stage if multiple statistics need to be analyzed. If needed, the PPCA can be extended to a mixture model to suit non-Gaussian distributed process data [20].

Based on the PPCA under NOC, we derive a statistical criterion to quantify the contribution of multiple missing variables. Using the criterion, when the occurrence of fault is identified through on-line process monitoring, the fault diagnosis can then be conducted by solving a series of subset selection problems. The combinatorial difficulty of the subset selection problem is tackled by a numerically efficient branch and bound (BAB) algorithm developed to isolate the set of faulty variables based on the criterion derived. The efficiency of the proposed method is demonstrated using a linear benchmark example [15], the Tennessee Eastman (TE) process [21] and randomly generated cases. It is worth to point out that although the proposed algorithm is developed based on the PPCA, the methodology is readily extendable to other MSPM methods.

The rest of this paper is organised as follows: after introducing the general principle of the PPCA, the methodology of missing variable analysis for fault isolation is presented in Section 2. Section 3 proposes the BAB algorithm for missing variable selection. The efficiency and effectiveness of the developed algorithms are demonstrated through various numerical examples in Section 4. Finally, the work is concluded in Section 5.

## 2 Probabilistic principal component analysis

Principal component analysis (PCA) [22] is a general multivariate statistical projection technique for dimension reduction. The central idea of PCA is to project the original $r$-dimensional data, $\mathbf{x}$, onto a space where the variance is maximized:

$$\mathbf{x} = \mathbf{W}\mathbf{t} + \bar{\mathbf{x}} + \mathbf{e} \tag{1}$$

where $\mathbf{W}$ refers to the eigenvectors of the sample covariance matrix corresponding to the $q$ ($q \leq r$) largest eigenvalues, $\mathbf{t}$ is the $q$-dimensional scores, $\bar{\mathbf{x}}$ is the mean of the data, and $\mathbf{e}$ is the noise term.

Recently, Tipping and Bishop [18] proposed a probabilistic formulation of PCA (PPCA) from the perspective of a Gaussian latent variable model. Specifically the noise is assumed to be Gaussian: $\mathbf{e} \sim G(\mathbf{0}, \sigma^2\mathbf{I})$, which implies

$$\mathbf{x}|\mathbf{t} \sim G(\mathbf{W}\mathbf{t} + \bar{\mathbf{x}}, \sigma^2\mathbf{I}) \tag{2}$$

Furthermore, by adopting a Gaussian distribution for the scores, $\mathbf{t} \sim G(\mathbf{0}, \mathbf{I})$, the marginal distribution of the data is also Gaussian: $\mathbf{x} \sim G(\bar{\mathbf{x}}, \mathbf{C})$, where the covariance matrix is $\mathbf{C} = \mathbf{W}\mathbf{W}^{\mathrm{T}} + \sigma^2\mathbf{I}$. In contrast to the sample covariance $\mathbf{S}$, the PPCA model defines $\mathbf{C}$ in terms of the auxiliary parameters $\mathbf{W}$ and $\sigma^2$. Note the number of free parameters in $\mathbf{C}$ is $rq + 1 - q(q-1)/2$, which is smaller than the $r(r+1)/2$ parameters in $\mathbf{S}$ if $q < r$ [18]. Therefore the PPCA provides a way to constrain the model complexity via the selection of $q$. The model parameters, $\{\bar{\mathbf{x}}, \mathbf{W}, \sigma^2\}$, can be estimated using the maximum likelihood algorithm; see [18] for details. Later, the PPCA was applied for process monitoring with improved fault detection capability [19].

The probabilistic framework of PPCA provides a single statistic for fault detection, as opposed to the $T^2$ and SPE in the traditional PCA. It was shown by Chen and Sun [17] that the data point $\mathbf{x}$ should be considered as out-of-control when

$$M^2 = (\mathbf{x} - \bar{\mathbf{x}})^{\mathrm{T}}\mathbf{C}^{-1}(\mathbf{x} - \bar{\mathbf{x}}) > \chi_r^2(\beta) \tag{3}$$

where $\chi_r^2(\beta)$ is the $\beta$-fractile of the chi-square distribution with $r$ degrees of freedom.

For the purpose of contribution analysis, Chen and Sun [17] suggested a missing variable approach in the PPCA framework. In particular, consider that the measurement vector $\mathbf{x}$ is partitioned into $n$-dimensional $\mathbf{x}_o$ and $d$-dimensional $\mathbf{x}_m$ as

$$\mathbf{x}^T = \begin{bmatrix} \mathbf{x}_o^T & \mathbf{x}_m^T \end{bmatrix} \tag{4}$$

where the subscripts $o$ and $m$ refer to observed and missing variables, respectively. Similarly, let the mean ($\bar{\mathbf{x}}$) and covariance matrix ($\mathbf{C}$) be partitioned as

$$\bar{\mathbf{x}} = \begin{bmatrix} \bar{\mathbf{x}}_o \\ \bar{\mathbf{x}}_m \end{bmatrix}; \quad \mathbf{C} = \begin{bmatrix} \mathbf{C}_{oo} & \mathbf{C}_{om} \\ \mathbf{C}_{mo} & \mathbf{C}_{mm} \end{bmatrix} \tag{5}$$

The conditional mean ($\bar{\mathbf{x}}_{m|o}$) and covariance matrix ($\mathbf{C}_{m|o}$) of $\mathbf{x}_m$ given $\mathbf{x}_o$ are

$$\bar{\mathbf{x}}_{m|o} = \bar{\mathbf{x}}_m + \mathbf{C}_{mo}\mathbf{C}_{oo}^{-1}(\mathbf{x}_o - \bar{\mathbf{x}}_o) \tag{6}$$

$$\mathbf{C}_{m|o} = \mathbf{C}_{mm} - \mathbf{C}_{mo}\mathbf{C}_{oo}^{-1}\mathbf{C}_{om} \tag{7}$$

Then, the conditional mean ($\bar{\mathbf{x}}_{|o}$) and covariance matrix ($\mathbf{C}_{|o}$) of whole $\mathbf{x}$ given $\mathbf{x}_o$ are

$$\bar{\mathbf{x}}_{|o} = \begin{bmatrix} \mathbf{x}_o \\ \bar{\mathbf{x}}_{m|o} \end{bmatrix}; \quad \mathbf{C}_{|o} = \begin{bmatrix} 0 & 0 \\ 0 & \mathbf{C}_{m|o} \end{bmatrix} \tag{8}$$

Therefore, the re-calculated monitoring statistic with missing values is given by

$$E[M^2] = \mathrm{tr}(\mathbf{C}^{-1}((\bar{\mathbf{x}}_{|o} - \bar{\mathbf{x}})(\bar{\mathbf{x}}_{|o} - \bar{\mathbf{x}})^T + \mathbf{C}_{|o})) \tag{9}$$

where $E[\cdot]$ denotes the expectation operator.

Now, the variable can be considered to be out-of-control if $E[M^2]$ in (9) exceeds $\chi_r^2(\beta)$. This criterion is applicable to any number of missing variables. In the original method of Chen and Sun [17], each individual variable of $\mathbf{x}$ is regarded as missing, and the monitoring statistic in (9) is re-calculated. If a variable contributes significantly to the data being detected as faulty, then the re-calculated statistic will be dramatically reduced. Furthermore, if $E[M^2]$ is smaller than the confidence bound, then we can say that by removing the corresponding variable and replacing it by its conditional mean, the process would return to the normal operating region. However, this approach is not capable of studying the joint contribution of multiple variables.

From the fault diagnosis point of view, the original source of a fault should correspond to a small set of measurements. Therefore, the objective becomes to select a minimum number of missing variables, whose re-calculated monitoring statistic is below the confidence bound. The flow of the method is given by the following algorithm:

**Algorithm 1** *Initially set* $d = 1$.

1. *Select d missing variables such that the re-calculated monitoring statistic is minimized.*

2. *If the minimum statistic is below the confidence bound, then the corresponding variables are isolated as the source of fault, and the algorithm can be terminated.*

3. *Otherwise, set $d = d + 1$, and return to Step 1.*

The selection of $d$-dimensional $\mathbf{x}_d$ from $r$-dimensional $\mathbf{x}$ in Step 1 of Algorithm 1 is a combinatorial optimization problem, which is NP-hard. One of the main contributions of this work is the development of a branch and bound (BAB) algorithm to solve this optimization problem efficiently, as discussed in the next section.

# 3 Branch and bound method

## 3.1 General principle

Let $X_r = \{x_i | i = 1, 2, \cdots, r\}$, be an $r$-element set. A subset selection problem with the selection criterion $\phi$ involves finding the optimal solution, $X_n^*$, such that

$$\phi(X_n^*) = \min_{X_n \subset X_r} \phi(X_n) \tag{10}$$

For this problem, the number of alternatives is $\mathcal{C}_r^n = \frac{r!}{(r-n)!n!}$, which grows very quickly with $r$ and $n$ rendering exhaustive search unviable. A BAB approach can provide globally optimal solution for the subset selection problem in (10) without exhaustive search. In this approach, the original problem (node) is divided (branched) into several non-overlapping subproblems (sub-nodes). If any of the $n$-element solution of a sub-problem cannot lead to the optimal solution, the sub-problem is not evaluated further (pruned), else it is branched again. The pruning of sub-problems allows the BAB approach to gain efficiency in comparison with exhaustive search.

The available BAB methods for subset selection can be classified as downwards [23–27] and upwards [28–30] BAB methods. The distinguishing feature of these approaches is the search direction. While the subset size is gradually decreased from $r$ to $n$ in a downwards BAB approach, it is gradually increased from 0 to $n$ in the upwards BAB approach. The use of downwards (upwards) BAB approach is appropriate, when the lower bound on the selection criteria holds for subset sizes larger (smaller) than the target subset size $n$ [28]. As shown in Section 3.2, the lower bound on the monitoring statistic in (9) holds in the upwards direction and thus the upwards BAB approach is used in this paper. In the subsequent discussion, we provide a brief overview of the upwards BAB approach; see [28] for details.

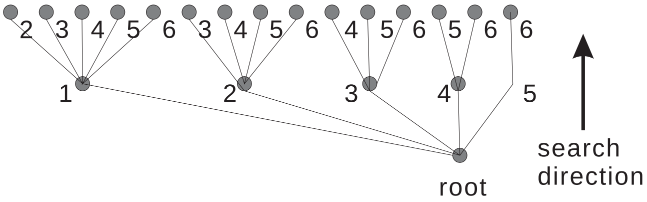


Figure 1: Solution tree for selecting 2 out of 6 elements

In an upwards BAB approach, each node has a fixed set $F_f$ and a candidate set $C_c$, which have $f$ and $c$ elements, respectively. The relationship between the fixed and candidate sets of a node and its $i^{\text{th}}$ sub-node (branching rule) is given as follows:

$$F^i_{f+1} = F_f \cup x_i; \quad C^i_{c-i} = C_c \setminus \{x_1, \cdots, x_i\} \tag{11}$$

where $F^i_{f+1}$ and $C^i_{c-i}$ denote the fixed and candidate sets of the $i^{\text{th}}$ sub-node and $i = 1, 2, \cdots, f + c - n + 1$. Based on (11), it can be noted that $F_f$ is gradually expanded using the elements of $C_c$, until the dimension of $F_f$ reaches the target subset size $n$. An example of the solution tree obtained by recursively applying the branching rule in (11) is shown in Figure 1. For the root node in this solution tree, we have $F_f = \emptyset$ and $C_c = X_r$. The label of the nodes denote the element being moved from $C_c$ to $F_f$. The solution tree has $\mathcal{C}^r_n$ terminal nodes, which represent different $n$-element subsets of $X_r$.

To describe the pruning principle, let $\mathcal{X}$ denote the ensemble of all $n$-element subsets, which can be obtained by expanding $F_f$ using (11), *i.e.*,

$$\mathcal{X} = \{\{F_f, X_{n-f}\} | X_{n-f} \subset C_c\} \tag{12}$$

and $\underline{\phi}(F_f)$ be the lower bound on $\phi$ computed over all elements of $\mathcal{X}$, *i.e.*

$$\underline{\phi}(F_f) = \min_{X_n \in \mathcal{X}} \phi(X_n) \tag{13}$$

Assume that $B$ is an upper bound of the globally optimal criterion, *i.e.* $B \geq \phi(X^*_n)$. Then,

$$\phi(X_n) > \phi(X^*_n) \quad \forall X_n \in \mathcal{X}, \quad \text{if} \;\; \underline{\phi}(F_f) > B \tag{14}$$

Hence, any $X_n \in \mathcal{X}$ cannot be optimal and can be pruned without further evaluation, if $\underline{\phi}(F_f) > B$.

Although pruning of nodes using (14) results in an efficient BAB algorithm, further efficiency can be gained by performing pruning on the sub-nodes directly. This happens as the lower bounds for different sub-nodes are related and can be computed together from $\underline{\phi}(F_f)$ resulting in computational efficiency. For $x_i \in C_c$, the $i^{\text{th}}$ sub-node can be pruned if

$$\underline{\phi}(F_f \cup x_i) > B \tag{15}$$

The pruning of $i^{\text{th}}$ sub-node is conducted by discarding $x_i$ from $C_c$, such that $x_i$ is not included in the fixed sets of any subsequent sub-nodes. If (15) is satisfied for multiple sub-nodes simultaneously, all these elements can be discarded from $C_c$ together.

For a BAB method involving pruning of sub-nodes, branching needs to be carried on sub-node level as well, which requires choosing a decision element to branch upon. Here, the decision element is selected as the element with smallest $\underline{\phi}(F_f \cup x_i)$ among all $x_i \in C_c$ (best-first search). The branching operation can be conducted by moving $x_i$ from $C_c$ to $F_f$ such that $x_i$ is included in the fixed sets of any subsequent sub-nodes.

## 3.2 Application to fault isolation problem

To apply the BAB approach for fault isolation purposes, we first express the monitoring statistic in (9) in a form that is more amenable to the application of BAB method. We have

$$E[M^2] = \text{tr}(\mathbf{C}^{-1}\mathbf{C}_{|o}) + (\bar{\mathbf{x}}_{|o} - \bar{\mathbf{x}})^T \mathbf{C}^{-1}(\bar{\mathbf{x}}_{|o} - \bar{\mathbf{x}}) \tag{16}$$

Using (8), $\text{tr}(\mathbf{C}^{-1}\mathbf{C}_{|o}) = \text{tr}([\mathbf{C}^{-1}]_{22}\mathbf{C}_{|o})$, where $[\mathbf{C}^{-1}]_{22}$ denotes the (2,2) block of $\mathbf{C}^{-1}$ having same dimensions as $\mathbf{C}_{|o}$. Based on formula for inverse of partitioned matrices [31], $[\mathbf{C}^{-1}]_{22} = (\mathbf{C}_{mm} - \mathbf{C}_{mo}\mathbf{C}_{oo}^{-1}\mathbf{C}_{om})^{-1} = \mathbf{C}_{m|o}^{-1}$. Therefore, $\text{tr}(\mathbf{C}^{-1}\mathbf{C}_{|o}) = d$, where $d$ is the number of missing variables given in Algorithm 1.

For the second term of (16), note that we can write

$$\bar{\mathbf{x}}_{|o} - \bar{\mathbf{x}} = \begin{bmatrix} \mathbf{x}_o - \bar{\mathbf{x}}_o \\ \mathbf{C}_{mo}\mathbf{C}_{oo}^{-1}(\mathbf{x}_o - \bar{\mathbf{x}}_o) \end{bmatrix} = \begin{bmatrix} \mathbf{C}_{oo} \\ \mathbf{C}_{mo} \end{bmatrix} \mathbf{C}_{oo}^{-1}(\mathbf{x}_o - \bar{\mathbf{x}}_o)$$

and

$$(\bar{\mathbf{x}}_{|o} - \bar{\mathbf{x}})^T \mathbf{C}^{-1} (\bar{\mathbf{x}}_{|o} - \bar{\mathbf{x}}) = (\mathbf{x}_o - \bar{\mathbf{x}}_o)^T \mathbf{C}_{oo}^{-1} \begin{bmatrix} \mathbf{C}_{oo}^T & \mathbf{C}_{mo}^T \end{bmatrix} \mathbf{C}^{-1} \begin{bmatrix} \mathbf{C}_{oo} \\ \mathbf{C}_{mo} \end{bmatrix} \mathbf{C}_{oo}^{-1} (\mathbf{x}_o - \bar{\mathbf{x}}_o)$$

Note that $\begin{bmatrix} \mathbf{C}_{oo}^T & \mathbf{C}_{mo}^T \end{bmatrix}$ denotes the first $n = r - d$ rows of $\mathbf{C}$, as $\mathbf{C}_{oo}$ is symmetric and $\mathbf{C}_{om} = \mathbf{C}_{mo}^T$. Thus,

$$\begin{aligned} (\bar{\mathbf{x}}_{|o} - \bar{\mathbf{x}})^T \mathbf{C}^{-1} (\bar{\mathbf{x}}_{|o} - \bar{\mathbf{x}}) &= (\mathbf{x}_o - \bar{\mathbf{x}}_o)^T \mathbf{C}_{oo}^{-1} \begin{bmatrix} \mathbf{I} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{C}_{oo} \\ \mathbf{C}_{mo} \end{bmatrix} \mathbf{C}_{oo}^{-1} (\mathbf{x}_o - \bar{\mathbf{x}}_o) \\ &= (\mathbf{x}_o - \bar{\mathbf{x}}_o)^T \mathbf{C}_{oo}^{-1} \mathbf{C}_{oo} \mathbf{C}_{oo}^{-1} (\mathbf{x}_o - \bar{\mathbf{x}}_o) \\ &= (\mathbf{x}_o - \bar{\mathbf{x}}_o)^T \mathbf{C}_{oo}^{-1} (\mathbf{x}_o - \bar{\mathbf{x}}_o) \end{aligned}$$

Finally, the simplified criterion is given as

$$E[M^2] = (\mathbf{x}_o - \bar{\mathbf{x}}_o)^T \mathbf{C}_{oo}^{-1} (\mathbf{x}_o - \bar{\mathbf{x}}_o) + d \tag{17}$$

During the application of Algorithm 1, $d = r - n$ is constant during every iteration. Thus, variable selection can be carried out by minimizing the first term in (17). Furthermore, $\bar{\mathbf{x}}_o$ can be subtracted in a preprocessing stage to simplify calculations. By defining $\mathbf{y} = \mathbf{x} - \bar{\mathbf{x}}$, the $n$ observed variables can be selected (equivalent to selecting $d$ missing variables) by solving the following problem:

$$\min_{X_n \subset X_r} \phi(X_n) = \mathbf{y}_{X_n}^T (\mathbf{C}_{X_n, X_n})^{-1} \mathbf{y}_{X_n} \tag{18}$$

where $X_r = \{1, 2, \cdots, r\}$, $\mathbf{y}_{X_n}$ denotes the elements of $\mathbf{y}$ with indices in $X_n$ and $\mathbf{C}_{X_n, X_n}$ represents the principal submatrix of $\mathbf{C}$ with rows and columns indexed by $X_n$.

The use of BAB for solving the optimization problem in (18) requires a lower bound on the selection criteria, calculated over all $\mathcal{X}$ in (12). This lower bound is derived in the next proposition.

**Proposition 1** Consider a node with fixed set $F_f$ and candidate set $C_c$. For $\mathcal{X}$ in (12),

$$\phi(F_f) \leq \min_{X_n \in \mathcal{X}} \phi(X_n) \tag{19}$$

*Proof*: Let $\mathbf{R}$ and $\tilde{\mathbf{R}}$ be the Cholesky factors of $\mathbf{C}_{X_n, X_n}$ for some $X_n \in \mathcal{X}$ and $\mathbf{C}_{F_f, F_f}$, respectively, *i.e.* $\mathbf{R}^T \mathbf{R} = \mathbf{C}$ and $\tilde{\mathbf{R}}^T \tilde{\mathbf{R}} = \mathbf{C}_{F_f, F_f}$. As $F_f \subset X_n$, $\tilde{\mathbf{R}}$ is a principal submatrix of $\mathbf{R}$, which implies that $\tilde{\mathbf{R}}^{-1}$ is a principal submatrix of $\mathbf{R}^{-1}$. Define $\mathbf{z} = \mathbf{R}^{-T} \mathbf{y}_{X_n}$. Then,

$$\phi(X_n) = \mathbf{y}_{X_n}^T \mathbf{R}^{-1} \mathbf{R}^{-T} \mathbf{y}_{X_n} = \mathbf{z}^T \mathbf{z} \tag{20}$$

Similarly,

$$\phi(F_f) = \mathbf{y}_{F_f}^T \tilde{\mathbf{R}}^{-1}\tilde{\mathbf{R}}^{-T}\mathbf{y}_{F_f} = \tilde{\mathbf{z}}^T\tilde{\mathbf{z}} \tag{21}$$

where $\tilde{\mathbf{z}} = \tilde{\mathbf{R}}^{-T}\mathbf{y}_{F_f}$. Since $\tilde{\mathbf{z}}$ is a subset of $\mathbf{z}$, $\phi(F_f) \leq \phi(X_n)$ and (19) follows. ■

Proposition 1 implies that the non-optimal nodes can be pruned based on $\phi(F_f)$. To gain further efficiency by pruning the sub-nodes directly, we relate the selection criteria of a node with its sub-nodes in the next proposition.

**Proposition 2** Consider a node with fixed set $F_f$ and candidate set $C_c$. For $x_i \in C_c$, $i = 1, 2, \cdots, c$,

$$\phi(F_f \cup i) = \phi(F_f) + \alpha_i \tag{22}$$

where

$$\alpha_i = \frac{\left(\mathbf{y}_i - \mathbf{y}_{F_f}\mathbf{C}_{F_f,F_f}^{-1}\mathbf{C}_{F_f,i}\right)^2}{(\mathbf{C}_{i,i} - \mathbf{C}_{F_f,i}^T\mathbf{C}_{F_f,F_f}^{-1}\mathbf{C}_{F_f,i})^2} \tag{23}$$

*Proof*: Let $\mathbf{Q}$ be the Cholesky factor of $\mathbf{C}_{(F_f\cup i),(F_f\cup i)}$, *i.e.* $\mathbf{Q}^T\mathbf{Q} = \mathbf{C}_{(F_f\cup i),(F_f\cup i)}$. Through simple algebraic manipulations, it can be shown that

$$\mathbf{Q} = \begin{bmatrix} \mathbf{R} & \mathbf{p}_i \\ 0 & \delta_i \end{bmatrix} \tag{24}$$

where $\mathbf{R}$ is the Cholesky factor of $\mathbf{C}_{F_f,F_f}$, $\mathbf{p}_i = \mathbf{R}^{-T}\mathbf{C}_{F_f,i}$ and $\delta_i = \sqrt{\mathbf{C}_{i,i} - \mathbf{p}_i^T\mathbf{p}_i}$. Using the formula for inversion of partitioned matrices [32], we get

$$\mathbf{Q}^{-1} = \begin{bmatrix} \mathbf{R}^{-1} & -\mathbf{R}^{-1}\mathbf{p}_i/\delta_i \\ 0 & \mathbf{1}/\delta_i \end{bmatrix} \tag{25}$$

Since, $\mathbf{y}_{F_f\cup i} = \begin{bmatrix} \mathbf{y}_{F_f} & \mathbf{y}_i \end{bmatrix}$,

$$\begin{aligned} \phi(F_f \cup i) &= \mathbf{y}_{F_f\cup i}^T\mathbf{Q}^{-1}\mathbf{Q}^{-T}\mathbf{y}_{F_f\cup i} && (26) \\ &= \mathbf{y}_{F_f}^T\mathbf{R}^{-1}\mathbf{R}^{-T}\mathbf{y}_{F_f} + \frac{1}{\delta_i^2}\left(\mathbf{y}_i - \mathbf{y}_{X_s}\mathbf{R}^{-1}\mathbf{p}_i\right)^2 && (27) \end{aligned}$$

Now, the result follows by noting that $\phi(F_f) = \mathbf{y}_{F_f}^T\mathbf{R}^{-1}\mathbf{R}^{-T}\mathbf{y}_{F_f}$ and substituting for $\mathbf{p}_i$ and $\delta_i$. ■

The evaluation of (22) requires inversion of only one matrix $\mathbf{C}_{F_f,F_f}$, which is the same for all $i \in C_c$. Thus, the use of (22) to obtain the selection criteria for all sub-nodes together is computationally more efficient than directly evaluating the selection criteria for every node. In summary, the following BAB algorithm can be used as Step 1 of Algorithm 1 for fault isolation purposes.

**Algorithm 2** *Initialize $f = 0$, $F_f = \emptyset$, $C_c = X_r$, $\phi(F_f) = 0$ and $B = \infty$. Call the following recursive algorithm:*

1. *If $\phi(F_f) > B$, prune the current node and return, else perform the following steps.*

2. *Calculate $\alpha_i$ in (23) $\forall i \in C_c$. Prune the subsets with $\phi(F_f) + \alpha_i > B$.*

3. *If $f = n$, go to next step. Otherwise, generate the $c$ sub-nodes according to the branching rule in (11) and call the recursive algorithm in Step 1 for each sub-node. Return to the caller after the execution of the loop finishes.*

4. *Find $J_{\min} = \phi(F_f) + \min_{i \in C_c} \alpha_i$. If $J_{\min} < B$, update $B = J_{\min}$. Return to the caller.*

# 4 Examples

In this section, the fault isolation approach (Algorithm 1) is applied to a linear benchmark example and the Tennessee Eastman benchmark problem. In addition, the computational efficiency of the proposed BAB algorithm (Algorithm 2) is further evaluated using randomly generated cases. All computations are carried out on a notebook with Intel® Core™ Duo Processor T2400 (1.83 GHz, 2MB RAM) using MATLAB® 2007b.

## 4.1 Linear benchmark example

To illustrate the principle of missing variable analysis for fault isolation and algorithms proposed in this work, the linear benchmark example presented in [15] is studied. The system has 5 sensor variables and 2 internal states, represented as follows:

$$\mathbf{x} = \mathbf{G}\mathbf{t} + \mathbf{e} \tag{28}$$

where

$$\mathbf{G} = \begin{bmatrix} -0.1670 & -0.1352 \\ -0.5671 & -0.3695 \\ -0.1608 & -0.1019 \\ 0.7574 & -0.0563 \\ -0.2258 & 0.9119 \end{bmatrix}$$

The internal state, $\mathbf{t}$ is distributed normally with zero mean and unit variance. Under NOC, the sensor error, $\mathbf{e}$ is normally distributed with zero mean and 0.01 variance. Firstly, 1000 samples representing NOC were produced through simulation. The covariance matrix, $\mathbf{C}$ of the PPCA model is as follows:

$$\mathbf{C} = \begin{bmatrix} 0.0604 & 0.1548 & 0.0435 & -0.1247 & -0.0983 \\ 0.1548 & 0.4963 & 0.1369 & -0.4270 & -0.2400 \\ 0.0435 & 0.1369 & 0.0491 & -0.1225 & -0.0634 \\ -0.1247 & -0.4270 & -0.1225 & 0.5997 & -0.2020 \\ -0.0983 & -0.2400 & -0.0634 & -0.2020 & 0.9262 \end{bmatrix} \tag{29}$$

The upper control limit of the $M^2$ statistic for a 95% confidence level is $\chi^2_5(0.95) = 11.07$.

Then, a single sensor fault with fault magnitude being 1.8 in the $4^{th}$ sensor is considered. After subtracting the mean calculated at NOC, the deviation of the faulty sensor vector is $\mathbf{y} = \begin{bmatrix} -0.079 & -0.59 & -0.22 & -1.78 & -0.024 \end{bmatrix}^T$. The $M^2$ statistic for this fault is $244.43 > 11.07$. Hence this fault is easily detected by the PPCA. To isolate the fault, the conventional contribution plot is shown in Figure 2. Due to the significant correlation between the second and the fourth sensors,

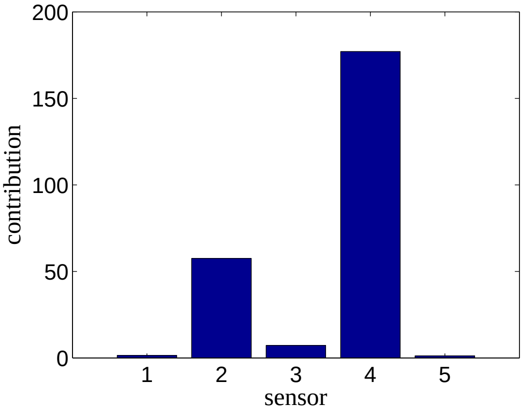


Figure 2: PPCA contribution plot for the single sensor fault.

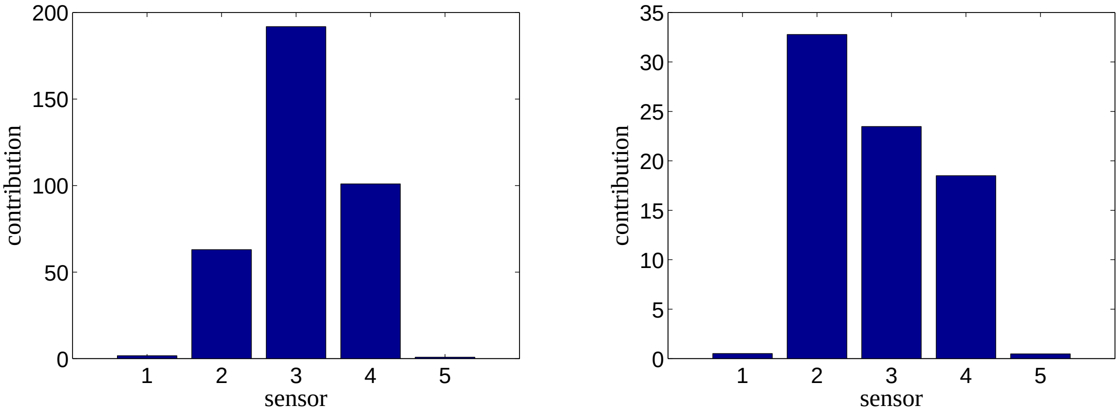
as indicated by Yue and Qin [15], the fault in the fourth sensor causes large contributions from both second and fourth sensors. Hence, the conventional contribution analysis will not be able to correctly isolate this single sensor fault.

To apply the missing variable analysis, according to Algorithm 1, the values of $E[M^2]$ for 5 cases, where one of the variables are considered to be missing are shown in Table 1. Table 1 clearly indicates that $E[M^2]$ is less than the upper control limit, only when $x_4$ is considered to be missing. Thus, the fourth sensor is correctly identified as the source of the fault by the PPCA.

Table 1: Missing variable analysis for the single sensor fault

| Missing Variable | $E[M^2]$ |
|---|---|
| $\{x_4\}$ | 3.02 |
| $\{x_5\}$ | 28.66 |
| $\{x_2\}$ | 67.15 |
| $\{x_1\}$ | 233.82 |
| $\{x_3\}$ | 241.73 |

Next, a multi-sensor fault with fault magnitude being 1.5 in both the third and fourth sensors is studied. The fault deviation vector is $\mathbf{y} = \begin{bmatrix} -0.079 & -0.59 & 1.49 & -1.48 & -0.024 \end{bmatrix}^T$. The corresponding monitoring criterion is $E[M^2] = 385.32 > 11.07$, which successfully detects the fault. For fault isolation, the PPCA contribution plots of 5 sensors are shown in Figure 3(a). The contribution plot incorrectly indicates that sensors 2, 3 and 4 are faulty due to the strong correlation between sensors 2 and 4.



(a) Fault Magnitude = 1.5

(b) Fault Magnitude = 0.5

Figure 3: PPCA contribution plot for the multi-sensor fault.

To apply Algorithm 1, first all cases with one of the variables missing are analysed. The minimum $E[M^2]$ for these cases is 145.38, which is larger than the upper control limit, indicating that the fault is not a single sensor fault. Further, all cases with two of the variables missing are evaluated and shown in Table 2. These results clearly and correctly indicate that sensors 3 and 4 are faulty.

Table 2: Missing variable analysis for the multi-sensor fault

| Fault magnitude = 1.5 | | Fault magnitude = 0.5 | |
|---|---|---|---|
| Missing Variables | $E[M^2]$ | Missing Variables | $E[M^2]$ |
| $\{x_3, x_4\}$ | 3.67 | $\{x_3, x_4\}$ | 3.67 |
| $\{x_3, x_5\}$ | 22.51 | $\{x_2, x_3\}$ | 6.23 |
| $\{x_2, x_3\}$ | 25.35 | $\{x_3, x_5\}$ | 8.68 |
| $\{x_1, x_2\}$ | 118.69 | $\{x_1, x_2\}$ | 14.77 |
| $\{x_2, x_4\}$ | 138.46 | $\{x_2, x_4\}$ | 19.39 |
| $\{x_2, x_5\}$ | 142.82 | $\{x_2, x_5\}$ | 19.68 |
| $\{x_1, x_3\}$ | 178.17 | $\{x_1, x_4\}$ | 41.25 |
| $\{x_4, x_5\}$ | 245.66 | $\{x_4, x_5\}$ | 41.28 |
| $\{x_1, x_4\}$ | 246.28 | $\{x_1, x_5\}$ | 43.22 |
| $\{x_1, x_5\}$ | 253.01 | $\{x_1, x_3\}$ | 52.98 |

Finally, to evaluate the effect of fault to noise ratio on the proposed algorithm, the case of multi-sensor fault is considered again with fault magnitude being 0.5. As the maximum value of noise for different sensors is approximately 0.31, the fault magnitude is comparable to the noise level. In this case, $E[M^2]$ is $75.74 > 11.07$ and thus the fault is successfully detected by the PPCA. Similar to the previous case, the contribution plots, shown in Figure 3(b), incorrectly identify faults in sensors 2, 3 and 4. For all cases with one of the variables missing, the minimum $E[M^2]$ is $18.77 > 11.07$. The values of $E[M^2]$ for all cases with two of the variables missing are shown in Table 2. Although $E[M^2]$ is lower than the control limit for multiple cases, the lowest value of $E[M^2]$ is still seen, when 3rd and 4th variables are considered to be missing.

This simple example illustrates how the missing variable analysis can be applied to isolate faulty variables and its advantages over traditional contribution plot based approach. Due to the strong correlation between sensors 2 and 4, both single sensor and multi-sensor faults were not correctly isolated by the contribution analysis. Nevertheless, the missing variable analysis successfully isolated all the faults.

### 4.2 Tennessee Eastman process

To demonstrate the effectiveness and efficiency, Algorithm 1 for missing variable analysis is applied for fault isolation of the Tennessee Eastman (TE) process [21]. This process has 5 main units, which are the reactor, condenser, separator, stripper and compressor. Streams of the plant consists of 8 components; A, B, C, D, E, F, G and H. Components A, B and C are gaseous reactants which are fed to the reactor to form products G and H. The flowsheet of the TE process is shown in Figure 4.

For fault isolation, the TE process is considered under closed-loop control as described by Downs and Vogel [21]. Data for the 41 available measurements and 11 out of 12 manipulated variables (MVs) are collected for 21 operational modes, which correspond to the normal and 20 faulty operation modes. These 52 measured variables are listed in Table 3. The root cause for five of these 20 faults are unknown according the original description of the plant [21]. Hence, it is not possible to validate the isolation results for these faults by analysing the process flowsheet. Furthermore, there are four faults, which are difficult to be detected by the PCA based approaches due to dynamic and nonlinear nature of these faults [7, 8]. Therefore, these 9 faults are excluded and the remaining 11 faults, listed in Table 4, are adopted for the case study. The identified variables for these eleven faults studied are listed in Table 5. Note that for some of the faults, Algorithm 1 identifies more than 1 set of variables, among which the variable set with smallest values of $E[M^2]$ is shown in Table 5.

As indicated in Table 5, most of the faults result in only one and two responsible variables. An exception is Fault 7, for which the BAB method indicates that there are up to seven measured variables responsive to this fault. To select 7 out of 52 measured variables, there are $C_{52}^{7} = 133,784,560$ alternatives. If one had to evaluate all alternatives to find the subset with minimum value of monitoring statistic, it would take more than a day to get the conclusion even if each evaluation takes only one millisecond. However, it only takes about 3.37 seconds for the BAB algorithm to find the contributing subset. This indicates that the proposed algorithm is suitable for on-line FDD to isolate contributing variables of a fault under investigation in a short time so that more sophisticated root-cause analysis can be carried out based on the isolated contributing variables and the consequent loss due to the fault can be minimised.

To better appreciate the proposed fault diagnosis algorithm, the relationship of the minimum criterion value $E[M^2]$ against the number of variables to be observed $n$ is shown in Figure 5. The horizontal line in Figure 5 is the upper control limit calculated as $\chi_r^2(\beta)$ with $r = 52$ and $\beta = 0.99$.

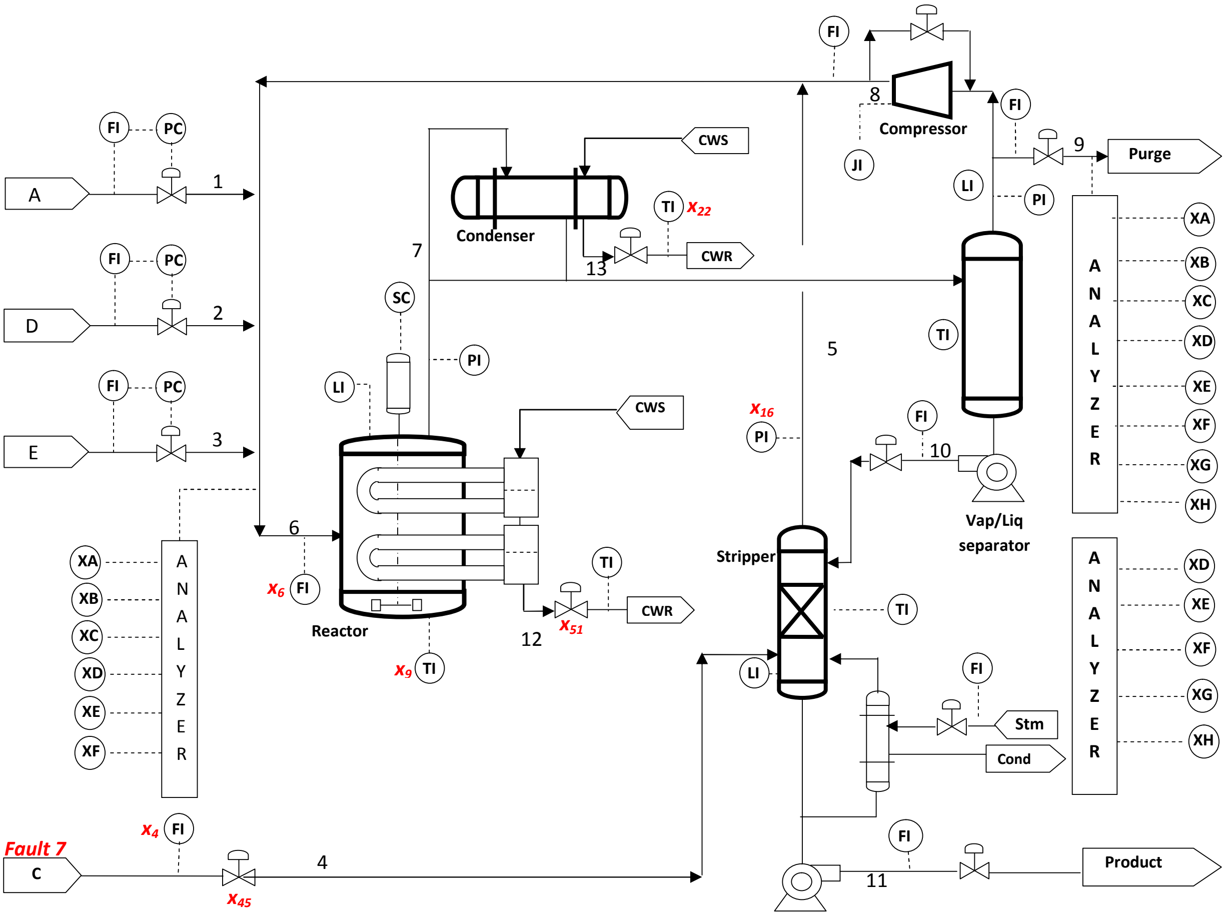


Figure 4: Flowsheet of the TEP Plant showing detected variables for Fault 7

Table 3: Measured variables

| ID | Description | ID | Description |
|---|---|---|---|
| $x_1$ | A Feed (Stream 1) | $x_{27}$ | Component E (Stream 6) |
| $x_2$ | D Feed (Stream 2) | $x_{28}$ | Component F (Stream 6) |
| $x_3$ | E Feed (Stream 3) | $x_{29}$ | Component A (Stream 9) |
| $x_4$ | Total Feed (Stream 4) | $x_{30}$ | Component B (Stream 9) |
| $x_5$ | Recycle Flow (Stream 8) | $x_{31}$ | Component C (Stream 9) |
| $x_6$ | Reactor Feed Rate (Stream 6) | $x_{32}$ | Component D (Stream 9) |
| $x_7$ | Reactor Pressure | $x_{33}$ | Component E (Stream 9) |
| $x_8$ | Reactor Level | $x_{34}$ | Component F (Stream 9) |
| $x_9$ | Reactor Temperature | $x_{35}$ | Component G (Stream 9) |
| $x_{10}$ | Purge Rate (Stream 9) | $x_{36}$ | Component H (Stream 9) |
| $x_{11}$ | Product Separator Temperature | $x_{37}$ | Component D (Stream 11) |
| $x_{12}$ | Product Separator Level | $x_{38}$ | Component E (Stream 11) |
| $x_{13}$ | Product Separator Pressure | $x_{39}$ | Component F (Stream 11) |
| $x_{14}$ | Product Separator Underflow (Stream 10) | $x_{40}$ | Component G (Stream 11) |
| $x_{15}$ | Stripper Level | $x_{41}$ | Component H (Stream 11) |
| $x_{16}$ | Stripper Pressure | $x_{42}$ | MV to D Feed Flow (Stream 2) |
| $x_{17}$ | Stripper Underflow (Stream 11) | $x_{43}$ | MV to E Feed Flow (Stream 3) |
| $x_{18}$ | Stripper Temperature | $x_{44}$ | MV to A Feed Flow (Stream 1) |
| $x_{19}$ | Stripper Steam Flow | $x_{45}$ | MV to Total Feed Flow (Stream 4) |
| $x_{20}$ | Compressor Work | $x_{46}$ | Compressor Recycle Valve |
| $x_{21}$ | Reactor Cooling Water Outlet Temperature | $x_{47}$ | Purge Valve (Stream 9) |
| $x_{22}$ | Separator Cooling Water Outlet Temperature | $x_{48}$ | MV to Separator Pot Liquid Flow (Stream 10) |
| $x_{23}$ | Component A (Stream 6) | $x_{49}$ | MV to Stripper Liquid Product Flow (Stream 11) |
| $x_{24}$ | Component B (Stream 6) | $x_{50}$ | Stripper Steam Valve |
| $x_{25}$ | Component C (Stream 6) | $x_{51}$ | MV to Reactor Cooling Water Flow |
| $x_{26}$ | Component D (Stream 6) | $x_{52}$ | MV to Condenser Cooling Water Flow |

Table 4: Operational Faults

| Fault ID | Description |
|---|---|
| 1 | Step in A/C Feed Ratio, B Composition Constant (stream 4) |
| 2 | Step in B composition while A/C ratio is constant (stream 4) |
| 4 | Step in Reactor Cooling Water Inlet Temperature |
| 5 | Step in Condenser Cooling Water Inlet Temperature |
| 6 | A Feed loss (step change in stream 1) |
| 7 | C Header Pressure Loss - reduced availability (step change in stream 4) |
| 8 | Random variation in A,B,C Feed Composition (stream 4) |
| 11 | Random variation in Reactor Cooling Water Inlet Temperature |
| 12 | Random variation in Condenser Cooling Water Inlet Temperature |
| 13 | Slow drift in Reaction Kinetics |
| 14 | Sticking Reaction Cooling Water Valve |

Table 5: Possible Fault responsive variables detected by the branch and bound algorithm

| Fault IDs | Variables eliminated |
|---|---|
| 1 | $\{x_{16}\}$ |
| 2 | $\{x_{21}, x_{24}, x_{30}\}$ |
| 4 | $\{x_9, x_{51}\}$ |
| 5 | $\{x_{11}, x_{22}\}$ |
| 6 | $\{x_1, x_{44}\}$ |
| 7 | $\{x_4, x_6, x_9, x_{16}, x_{22}, x_{45}, x_{51}\}$ |
| 8 | $\{x_{37}\}$ |
| 11 | $\{x_{51}\}$ |
| 12 | $\{x_{22}\}$ |
| 13 | $\{x_{37}\}$ |
| 14 | $\{x_9, x_{51}\}$ |

It is shown that for $n \leq 45$ (left to the dashed line), the minimum criterion value is less than the upper control limit, whilst for $n \geq 46$, (right to the dotted line) the criterion is above the upper control limit. Therefore, the maximum number of non-responsive variables is 45 and the number of possible responsive variables to Fault 7 is $52 - 45 = 7$. The actual deviations of these 7 variables are $y_4 = -14.5015$, $y_6 = -3.8761$, $y_9 = -3.1665$, $y_{16} = -4.2085$, $y_{22} = -3.4847$, $y_{45} = 4.4334$, and $y_{51} = -4.4255$. The corresponding locations of these 7 variables and the fault are marked in Figure 4.

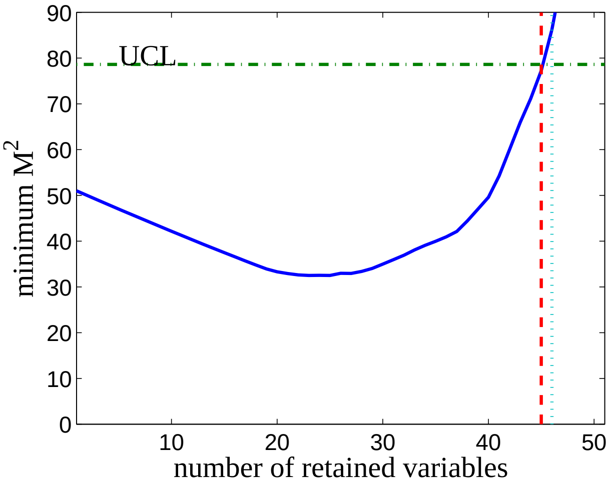


Figure 5: The minimum $M^2$ against number of variables (n) observed for Fault 7.

A physical explanation of the propagation of fault to the identified variables is as follows: Fault 7 shown in Figure 4 involves C Header Pressure Loss - reduced availability (step change in stream 4), resulting in a decrease in the Total Feed in stream 4 ($x_4$) and a corresponding increase in the MV to Total Feed Flow in stream 4 ($x_{45}$) to counter the effect of the fault occurrence through the corresponding flow control loop of the plant. The decrease in the Total Feed in stream 4 ($x_4$) results in a corresponding decrease in the Recycle back to the reactor through stream 6, thereby reducing the Reactor Feed Rate in stream 6 ($x_6$) as well as the Reactor Temperature ($x_9$). This decrease in the Reactor Temperature ($x_9$) results in a decrease in the Reactor Cooling Water Flow ($x_{51}$). Also, the decrease in the Total Feed in stream 4 ($x_4$) into the Stripper results in a decrease in the Stripper Pressure ($x_{16}$) and consequently, a decrease in the Separator Cooling Water Outlet Temperature ($x_{22}$). From the above explanation, it can be concluded that the results obtained through the BAB algorithm is also supported by the analysis through a physical understanding of the plant. Therefore, this proposed approach will be practically useful for fault isolation associating with condition monitoring.

It is worth to note that there are two pairs of faults, *i.e.* Faults 4 and 14, and Faults 8 and 13, which have the same faulty variable set isolated by the missing variable analysis. That the same faulty variable set is identified for Faults 4 and 14 is understandable as both of these faults are associated with the reactor cooling water although one is a step change in the cooling water temperature whilst another is due to a sticking cooling water control valve. The identification of same faulty variable set for Faults 8 and 13, however, is difficult to explain without the knowledge of the reaction kinetics, which was hidden for the benchmark problem [21]. Note that Algorithm 2 identifies several single-variables sets for these two faults in addition to the common set corresponding to the minimum $E[M^2]$. For Fault 8, three single-variable sets, $\{x_{37}\}$, $\{x_6\}$ and $\{x_{30}\}$ are identified, whilst for Fault 13, 8 single-variable sets are detected with $E[M^2] < \chi_r^2(\beta)$, which are $\{x_{37}\}$, $\{x_8\}$, $\{x_{30}\}$, $\{x_{26}\}$, $\{x_{40}\}$, $\{x_{23}\}$, $\{x_{22}\}$ and $\{x_{39}\}$. This extra information can provide useful insight for plant engineers to correctly diagnose the underlying causes of these two faults.

### 4.3 Random tests

To further evaluate the computational efficiency of the BAB in Algorithm 2, random cases are considered. The test involves retaining $n$ out of 40 variables. For each $n$, a vector $\mathbf{y}$ of dimension 40 and a positive definite matrix $\mathbf{C}$ of dimension $40 \times 40$ are randomly generated. To ensure positive-definiteness of $\mathbf{C}$, a matrix $\tilde{\mathbf{C}}$ is randomly generated, where $\mathbf{C} = \tilde{\mathbf{C}}\tilde{\mathbf{C}}^T$. The elements of $\mathbf{y}$ and $\tilde{\mathbf{C}}$ are normally distributed with zero means and unit variance. For each problem, 1000 cases are tested and the average computation time and number of nodes evaluated are shown in Figure 6. For comparison, the computational time of a brute force search is estimated by multiplying $\mathcal{C}_{40}^n$ with the time required for evaluating the criterion given in (17) of $n$ variables.

Figure 6 indicates that the worst efficiency of the BAB algorithm occurs for $n = 28$ instead of $n = 20$ as expected for the brute force search. This efficiency shift towards a higher number of variables is due to the upwards nature of the BAB algorithm. Further explanation of the efficiency shift for different types of BAB algorithms can be found in Cao and Kariwala [28]. For most fault isolation applications, the number of missing variables to be detected is relatively small, *i.e.* the number of retained variables is relatively large. Thus, the efficiency shift indicated in Figure 6 is not favourable for fault isolation applications. Nevertheless, even for the worst case of $n = 28$, the average number of nodes evaluated and average solution time required by the BAB approach are still 5 orders of magnitude lower than that required by the brute-force search showing the advantages of the proposed

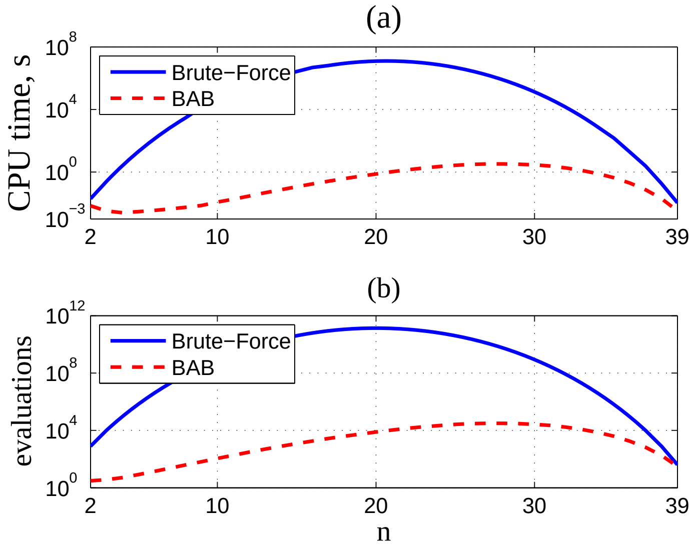


Figure 6: Random tests for retaining $n$ from 40 variables (a) computation time and; (b) number of nodes evaluated

approach. This random test together with the TE case study indicates that the proposed the BAB algorithm is capable of efficiently solving multivariate problems with up to 50 variables. In future, a bidirectional BAB algorithm [28–30] will be developed for handling fault isolation applications with more variables.

## 5 Conclusions

In contrast to considerable attention paid to multivariate statistical process monitoring (MSPM), research in fault diagnosis has been relatively limited due to the difficulty in dealing with the multivariate contribution analysis. It is the first time that a multivariate missing-variable approach and an efficient branch and bound algorithm are proposed to efficiently solve this problem. Although the focus of this paper is on using the probabilistic principal component analysis (PPCA), the framework developed in this work can be easily extended to adopt other criteria for fault diagnosis. The numerical case studies using a linear example and the Tennessee Eastman process show that the proposed method is able to find the minimum set of variables, which are affected by the fault, in a short time. This computational efficiency, which is examined through a random test, can give operators more time to identify and further deal with the fault in order to minimize the consequent loss due to the occurrence of the fault. However, the random tests also indicate that the largest size of variable set solvable by the BAB algorithm is limited. Future work will focus on improving the computational efficiency of the BAB algorithm to make it applicable to dynamic data with a large number of variables.

## 6 Acknowledgement

This work was supported by the Royal Society, UK through grant no. IV0871568.

## References

[1] V. Venkatasubramanian, R. Rengaswamy, K. Yin, and S.N. Kavuri. A review of process fault detection and diagnosis Part I: Quantitative model-based methods. *Computers and chemical engineering*, 27(3):293–311, 2003.

[2] S. Joe Qin. Statistical process monitoring: basics and beyond. *Journal of Chemometrics*, 17 (8-9):480–502, 2003.

[3] V. Venkatasubramanian, R. Rengaswamy, S.N. Kavuri, and K. Yin. A review of process fault detection and diagnosis Part III: Process history based methods. *Computers and Chemical Engineering*, 27(3):327–346, 2003.

[4] SI Alabi, AJ Morris, and EB Martin. On-line dynamic process monitoring using wavelet-based generic dissimilarity measure. *Chemical Engineering Research and Design*, 83(6):698–705, 2005.

[5] U. Kruger, Y. Zhou, and G.W. Irwin. Improved principal component monitoring of large-scale processes. *Journal of Process Control*, 14(8):879–888, 2004.

[6] J.M. Lee, C.K. Yoo, and I.B. Lee. Statistical monitoring of dynamic processes based on dynamic independent component analysis. *Chemical engineering science*, 59(14):2995–3006, 2004.

[7] P.P. Odiowei and Y. Cao. Nonlinear dynamic process monitoring using canonical variate analysis and kernel density estimations. *IEEE Transactions on Industrial Informatics*, 6(1):36–45, 2010.

[8] P.P. Odiowei and Y. Cao. State-space independent component analysis for nonlinear dynamic process monitoring. *Chemometrics and Intelligent Laboratory Systems*, In Press, Corrected Proof, 2010. ISSN 0169-7439. doi: DOI: 10.1016/j.chemolab.2010.05.014.

[9] M. N. Nounou, B. R. Bakshi, P. K. Goel, and X. Shen. Bayesian principal component analysis. *Journal of Chemometrics*, 16:576–595, 2002.

[10] T. Chen, J. Morris, and E. Martin. Probability density estimation via an infinite Gaussian mixture model: application to statistical process monitoring. *Journal of the Royal Statistical Society C (Applied Statistics)*, 55:699–715, 2006.

[11] S. W. Choi, J. H. Park, and I.-B. Lee. Process monitoring using a Gaussian mixture model via principal component analysis and discriminant analysis. *Computers and Chemical Engineering*, 28:1377–1387, 2004.

[12] U. Thissen, H. Swierenga, A. P. de Weijer, R. Wehrens, W. J. Melssen, and L. M. C. Buydens. Multivariate statistical process control using mixture modelling. *Journal of Chemometrics*, 19: 23–31, 2005.

[13] J. Yu and S. J. Qin. Multimode process monitoring with Bayesian inference-based finite Gaussian mixture models. *AIChE Journal*, 54:1811–1829, 2008.

[14] P. Miller, R. E. Swanson, and C. F. Heckler. Contribution plots: a missing link in multivariate quality control. *International Journal of Applied Mathematics and Computer Science*, 8:775–792, 1998.

[15] H.H. Yue and S.J. Qin. Reconstruction based fault identification using a combined index. *Industrial and Engineering Chemistry Research*, 40:4403–4414, 2001.

[16] Q. Chen, U. Kruger, M. Meronk, and A. Y. T. Leung. Synthesis of $T^2$ and $Q$ statistics for process monitoring. *Control Engineering Practice*, 12:745–755, 2004.

[17] T. Chen and Y. Sun. Probabilistic contribution analysis for statistical process monitoring: A missing variable approach. *Control Engineering Practice*, 17(4):469–477, 2009.

[18] M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society B*, 61:611–622, 1999.

[19] D. Kim and I.-B. Lee. Process monitoring based on probabilistic PCA. *Chemometrics and intelligent laboratory systems*, 67:109–123, 2003.

[20] S. W. Choi, E. B. Martin, and A. J. Morris. Fault detection based on a maximum-likelihood principal component analysis (PCA) mixture. *Industrial and Engineering Chemistry Research*, 44:2316–2327, 2005.

[21] J. J. Downs and E. F. Vogel. A plant-wide industrial process control problem. *Comput. Chem. Eng.*, 17(3):245–255, 1993.

[22] I. T. Jolliffe. *Principal Component Analysis.* Springer, 2nd edition, 2002.

[23] P. Narendra and K. Fukunaga. A branch and bound algorithm for feature subset selection. *IEEE Trans. Comput.*, C-26:917–922, 1977.

[24] B. Yu and B. Yuan. A more efficient branch and bound algorithm for feature selection. *Pattern Recognition*, 26:883–889, 1993.

[25] P. Somol, P. Pudil, F. Ferri, and J. Kittler. Fast branch and bound algorithm in feature selection. In B. Sanchez, J. Pineda, J. Wolfmann, Z. Bellahsense, and F. Ferri, editors, *Proceedings of World Multiconference on Systemics, Cybernetics and Informatics*, volume VII, pages 1646–651, Orlando, Florida, USA, 2000.

[26] X.-W. Chen. An improved branch and bound algorithm for feature selection. *Pattern Recognition Letters*, 24:1925–1933, 2003.

[27] Y. Cao and P. Saha. Improved branch and bound method for control structure screening. *Chem. Engg. Sci.*, 60(6):1555–1564, 2005.

[28] Y. Cao and V. Kariwala. Bidirectional branch and bound for controlled variable selection: Part I. Principles and minimum singular value criterion. *Comput. Chem. Engng.*, 32(10):2306–2319, 2008.

[29] V. Kariwala and Y. Cao. Bidirectional branch and bound for controlled variable selection: Part II. Exact local method for self-optimizing control. *Comput. Chem. Eng.*, 33:1402–1412, 2009.

[30] V. Kariwala and Y. Cao. Bidirectional branch and bound for controlled variable selection: Part III. Local average loss minimization. *IEEE Transactions on Industrial Informatics*, to appear, 2010.

[31] G. H. Golub and C. F. van Loan. *Matrix Computations.* The Johns Hopkins University Press, Baltimore, MD, 3rd edition, 1993.

[32] R. A. Horn and C. R. Johnson. *Matrix Analysis.* Cambridge University Press, Cambridge, UK, 1985.