2018 26th Mediterranean Conference on Control and Automation (MED)
Zadar, Croatia, June 19-22, 2018

Sparse reconstruction-based contribution for Multiple Fault Isolation by KPCA

Gilles Mourot[1], Maya Kallas[1], Kwami Anani[1], Didier Maquin[1]

Abstract— **This paper addresses the problem of multiple fault isolation based on kernel principal component analysis and proposes a sparse fault estimation method to evaluate the reconstruction-based contribution. The fault magnitude estimation is here formulated as an optimization problem under nonnegativity and sum-to-one constraints. A multiplicative iterative scheme and its initialization procedure are proposed to solve it. The effectiveness of the proposed method is demonstrated on the simulated continuous stirred tank reactor (CSTR) process.**

I. Introduction

Kernel Principal Component Analysis (KPCA) is one of the most popular methods for fault detection of nonlinear systems since the work of [1] and [2]. KPCA maps measurements from their original space into a higher dimensional feature space where principal component analysis (PCA) is performed. Fault detection is then performed in this feature space through straightforward extensions of PCA detection indices.

Fault isolation has been less addressed in the literature due to complexity related to the nonlinear mapping. Among the popular contribution methods, [3] proposed a reconstruction-based contribution (RBC) method for KPCA. They defined the estimation of fault magnitude minimizing the fault detection index along a presumed faulty direction as the contribution of that variable. However, their method suffers from three main shortcomings. Firstly, they only addressed single fault isolation. Secondly, due to nonlinear mapping, the RBC is not the estimated fault but the difference between the detection indices of the faulty measurement and its reconstruction. Finally, to solve this nonlinear optimization problem, they proposed an iterative fixed-point algorithm. In fact, this mapping back from the feature space into the input space is a pre-image problem which deals with a wider multivariate estimation issue. Furthermore, solving the pre-image problem, known as being ill-posed, can be tricky. For example, among the methods proposed to solve it, an iterative fixed-point scheme similar to that proposed by [3] was shown to be unstable and suffer from local minima [4]. A solution consists of minimizing the detection index subject to constraints or with additional penalty terms that are related to these constraints.

Addressing the more general issue of multiple fault isolation, we show that the estimation of fault's magnitude can be written as a linear combination of the difference between training data and faulty measurement along the reconstruction directions, the coefficients being training data contributions to the estimation. Thus positive and negative coefficients can balance in the linear combination taking into account they are normalized to sum to one. Therefore a sparse solution should be obtained by minimizing the number of coefficients different from zero. As an initial approach, ℓ_0 or ℓ_1 regularization function could be added as penalty in order to control the sparsity degree of solution. However, the detection index is already nonlinear with respect to fault magnitude, thus, this approach yields a computationally expensive problem and moreover the balance between regularization and estimation is often difficult to tune. Since sparse solution is more likely to arise in problems with nonnegativity constraints [5], [6], the fault magnitude estimation is then formulated as an optimization problem under nonnegativity and sum-to-one constraints on the coefficients.

To solve this fully constrained optimization problem, popular approaches are presented in literature, such as projected-gradient type methods and their extensions [7] and multiplicative update methods [5], [8]. At each iteration, the former essentially evaluates the gradient of the objective function and the projection onto the unit simplex defined by the constraints. Its main difficulty is the selection of the step-size to ensure the algorithm's convergence without constraint violations. The latter differs from projected-gradient methods because the constraints are involved in the update scheme in order to ensure that they are satisfied at each iteration. In the following, a multiplicative update algorithm based on the split gradient method [8] is proposed to solve the fully constrained optimization issue presented above.

The paper is organized as follows: after a short recall of fault detection by kernel principal component analysis in section II, multiple fault isolation by reconstruction-based contribution method is presented and the lack of sparsity of the solution is highlighted in section III-A. This fault magnitude estimation is then reformulated as an optimization problem under nonnegativity and sum-to-one constraints and a multiplicative iterative scheme and its initialization procedure are proposed to solve it in section III-B. Finally, in section IV, the proposed method is applied to a commonly used benchmark : a non-isothermal continuous chemical reactor on which different faults are simulated [9].

II. Fault Detection with KPCA

In this section, we briefly describe KPCA and define its use for Fault Detection. KPCA maps a dataset from the input

[1]Centre de Recherche en Automatique de Nancy, CNRS, Université de Lorraine 2, Avenue de la Forêt de Haye TSA 60604 54 518 Vandoeuvre-lès-Nancy, FRANCE (gilles.mourot, maya.kallas, didier.maquin)@univ-lorraine.fr

space $\mathcal{X}$ onto a high dimensional feature space $\mathcal{H}$ using a nonlinear function $\varphi(\cdot)$. The transformed data obtained in this feature space are then analyzed using linear PCA.

A. Kernel PCA

Let us consider a data matrix defined by n measurement vectors $\boldsymbol{x}_i \in \mathcal{X} \subseteq \mathbb{R}^m$ collected on the system under normal operation:

$$\mathbf{X} = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n]^\top \in \mathbb{R}^{n \times m}$$

Let $\varphi(\cdot)$ denotes the nonlinear mapping function from the input space $\mathcal{X}$ into the feature space $\mathcal{H}$:

$$\begin{aligned} \varphi\colon \mathcal{X} &\mapsto \mathcal{H} \\ \boldsymbol{x}_i &\mapsto \boldsymbol{\varphi}_i = \varphi(\boldsymbol{x}_i) \in \mathbb{R}^h \end{aligned}$$

where h is the dimension of the feature space.

The matrix gathering the mapped vectors are writen as follows:

$$\boldsymbol{\Phi} = [\boldsymbol{\varphi}_1, \cdots, \boldsymbol{\varphi}_n]^\top \in \mathbb{R}^{n \times h}$$

and its columns are assumed to be centered in the following.

KPCA is performed in the feature space $\mathcal{H}$ by diagonalizing the empirical covariance matrix:

$$\mathbf{S} = \frac{1}{n-1} \boldsymbol{\Phi}^\top \boldsymbol{\Phi}$$

The primal formulation of KPCA in the feature space is an eigenvalue/eigenvector problem:

$$\frac{1}{n-1} \boldsymbol{\Phi}^\top \boldsymbol{\Phi}\, \boldsymbol{v}_i = \lambda_i\, \boldsymbol{v}_i \qquad i = 1, \ldots, n$$

where $\boldsymbol{v}_i$ and $\lambda_i\, (i = 1, \ldots, n)$ are the eigenvectors and eigenvalues of the covariance matrix $\mathbf{S}$.

However, $\varphi(\cdot)$ does not need to be explicitly defined, the Gram matrix $\mathbf{K} = \boldsymbol{\Phi}\ \boldsymbol{\Phi}^\top$ is then evaluated from the kernel $\kappa\colon \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ fulfilling Mercer's conditions by:

$$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \varphi(\boldsymbol{x}_i)^\top \varphi(\boldsymbol{x}_j)$$

Indeed, the covariance matrix $\mathbf{S}$ and the matrix $\frac{1}{n-1}\mathbf{K}$ have the same non-zero eigenvalues and their eigenvectors are related by:

$$\mathbf{V} = \boldsymbol{\Phi}^\top \mathbf{A}$$

with

$$\mathbf{A} = \frac{1}{\sqrt{n-1}} \mathbf{U}\, \boldsymbol{\Lambda}^{-1/2}$$

and $r = \min(n, h)$, $\mathbf{V} = [\boldsymbol{v}_1, \cdots, \boldsymbol{v}_r] \in \mathbb{R}^{h \times r}$, $\mathbf{U} = [\boldsymbol{u}_1, \cdots, \boldsymbol{u}_r] \in \mathbb{R}^{n \times r}$ are the matrices of the eigenvectors associated to the diagonal matrix of eigenvalues $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1 \ldots \lambda_r)$.

By choosing a number ℓ of principal components, the feature space is decomposed into the principal and residual subspaces, spanned respectively by $\hat{\mathbf{V}}$ (the ℓ first eigenvectors of $\mathbf{V}$ corresponding to the ℓ largest eigenvalues) and $\tilde{\mathbf{V}}$ (the $r - \ell$ last eigenvectors of $\mathbf{V}$). The same partitioning is considered for $\mathbf{U}$ and $\boldsymbol{\Lambda}$.

In the following, the Gaussian kernel is used:

$$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\frac{(\boldsymbol{x}_i - \boldsymbol{x}_j)^T (\boldsymbol{x}_i - \boldsymbol{x}_j)}{2\, c}\right) \qquad (1)$$

where c is the kernel dispersion parameter.

B. Fault detection

After the KPCA model has been built, we now examine its use for sensor fault detection.

The detection index SPE (squared prediction error) is usually used to detect fault in the residual space since it measures the lack of fit of the mapped data to the KPCA model. Since the projection of $\boldsymbol{x}$ onto the residual subspace $\tilde{\mathbf{V}}$ is evaluated as:

$$\tilde{\boldsymbol{t}}(\boldsymbol{x}) = \tilde{\mathbf{V}}^\top \varphi(\boldsymbol{x})$$

the detection index SPE is given by:

$$\begin{aligned} SPE(\boldsymbol{x}) &= \tilde{\boldsymbol{t}}^\top(\boldsymbol{x}) \tilde{\boldsymbol{t}}(\boldsymbol{x}) \\ &= 1 - \boldsymbol{\kappa}^\top(\boldsymbol{x})\, \hat{\mathbf{Q}}\, \boldsymbol{\kappa}(\boldsymbol{x}) \end{aligned}$$

where

$$\begin{aligned} \hat{\mathbf{Q}} &= \hat{\mathbf{A}}\, \hat{\mathbf{A}}^\top \\ \boldsymbol{\kappa}(\boldsymbol{x}) &= [\kappa(\boldsymbol{x}_1, \boldsymbol{x}), \cdots, \kappa(\boldsymbol{x}_n, \boldsymbol{x})]^\top \end{aligned}$$

Abnormal situation in the data $\boldsymbol{x}$ is detected when $SPE(\boldsymbol{x})$ is greater than a threshold δ^2.

1) Remark.: The tuning of KPCA parameters goes beyond the scope of this paper. Interested readers should consult the extensive literature on the subject.

III. Sparse reconstruction-based contribution

The reconstructed directions are assumed to be known a priori. Then the reconstruction-based contribution for these different subsets of variables is evaluated and the variables belonging to the subset with the maximum RBC are considered as faulty.

A. Position of the problem

Let us show how to calculate the RBC for a subset R of supposed faulty variables. The reconstructed observation is defined by:

$$\boldsymbol{z}_\mathrm{R} = \boldsymbol{x} - \boldsymbol{\Xi}_\mathrm{R}\, \mathbf{f}_\mathrm{R} \qquad (2)$$

where $\boldsymbol{\Xi}_\mathrm{R}$ is the matrix of fault directions with 1 to indicate the faulty variables and 0 for the other variables.

The estimation of fault magnitudes is obtained by minimizing the detection index SPE:

$$\hat{\mathbf{f}}_\mathrm{R} = \arg\min_{\mathbf{f}_\mathrm{R}} SPE(\boldsymbol{z}_\mathrm{R}) \qquad (3)$$

Then the reconstruction-based contribution of subset R is defined by:

$$RBC_\mathrm{R} = SPE(\boldsymbol{x}) - SPE(\hat{\boldsymbol{z}}_\mathrm{R}) \qquad (4)$$

where $\hat{\boldsymbol{z}}_\mathrm{R}$ is the reconstructed observation obtained by replacing $\hat{\mathbf{f}}_\mathrm{R}$ in (2).

To solve the minimization problem (3), $SPE(\boldsymbol{z}_{\mathrm{R}})$ is derived with respect to vector $\mathbf{f}_{\mathrm{R}}$:

$$\frac{\partial SPE(\boldsymbol{z}_{\mathrm{R}})}{\partial \mathbf{f}_{\mathrm{R}}} = -2\,\frac{\partial \boldsymbol{\kappa}^{\top}(\boldsymbol{z}_{\mathrm{R}})}{\partial \mathbf{f}_{\mathrm{R}}}\,\hat{\mathbf{Q}}\boldsymbol{\kappa}(\boldsymbol{z}_{\mathrm{R}})$$

Let us evaluate the component-wise derivative of $\boldsymbol{\kappa}(\boldsymbol{z}_{\mathrm{R}})$ with respect to vector $\mathbf{f}_{\mathrm{R}}$:

$$\frac{\partial \kappa(\boldsymbol{z}_{\mathrm{R}}, \boldsymbol{x}_j)}{\partial \mathbf{f}_{\mathrm{R}}} = \frac{\partial \boldsymbol{z}_{\mathrm{R}}^{\top}}{\partial \mathbf{f}_{\mathrm{R}}}\,\frac{\partial \kappa(\boldsymbol{z}_{\mathrm{R}}, \boldsymbol{x}_j)}{\partial \boldsymbol{z}_{\mathrm{R}}}$$

From (2) and the gaussian kernel, we obtain:

$$\frac{\partial \kappa(\boldsymbol{z}_{\mathrm{R}}, \boldsymbol{x}_j)}{\partial \mathbf{f}_{\mathrm{R}}} = \frac{1}{c}\,\kappa(\boldsymbol{z}_{\mathrm{R}}, \boldsymbol{x}_j)\,\boldsymbol{\Xi}_{\mathrm{R}}^{\top}\,(\boldsymbol{z}_{\mathrm{R}} - \boldsymbol{x}_j)$$

Finally, the gradient can be written as:

$$\frac{\partial SPE(\boldsymbol{z}_{\mathrm{R}})}{\partial \mathbf{f}_{\mathrm{R}}} = -\frac{2}{c}\sum_{j=1}^{n}\beta_j\,\boldsymbol{\Xi}_{\mathrm{R}}^{\top}(\boldsymbol{z}_{\mathrm{R}} - \boldsymbol{x}_j) \tag{5}$$

with

$$\beta_j = \kappa(\boldsymbol{z}_{\mathrm{R}}, \boldsymbol{x}_j)\,\boldsymbol{\xi}_j^{\top}\,\hat{\mathbf{Q}}\,\boldsymbol{\kappa}(\boldsymbol{z}_{\mathrm{R}}) \tag{6}$$

and $\boldsymbol{\xi}_j$ is the j^{th} column of the identity matrix.

Let us replace $\boldsymbol{z}_{\mathrm{R}}$ (2) in (5) and set the gradient to zero to obtain the first order stationary condition. Taking into account the following necessary condition:

$$\hat{\mathbf{A}}^{\top}\,\boldsymbol{\kappa}(\hat{\boldsymbol{z}}_{\mathrm{R}}) \neq 0,$$

the solution is thus given by:

$$\hat{\mathbf{f}}_{\mathrm{R}} = \sum_{j=1}^{n}\hat{\alpha}_j\,\boldsymbol{\Xi}_{\mathrm{R}}^{\top}\,(\boldsymbol{x} - \boldsymbol{x}_j)$$

with

$$\hat{\alpha}_j = \frac{\kappa(\hat{\boldsymbol{z}}_{\mathrm{R}}, \boldsymbol{x}_j)\,\boldsymbol{\xi}_j^{\top}\,\hat{\mathbf{Q}}\,\boldsymbol{\kappa}(\hat{\boldsymbol{z}}_{\mathrm{R}})}{\sum_{t=1}^{n}\kappa(\hat{\boldsymbol{z}}_{\mathrm{R}}, \boldsymbol{x}_t)\,\boldsymbol{\xi}_j^{\top}\,\hat{\mathbf{Q}}\,\boldsymbol{\kappa}(\hat{\boldsymbol{z}}_{\mathrm{R}})}$$
$$\hat{\boldsymbol{z}}_{\mathrm{R}} = \boldsymbol{x} - \boldsymbol{\Xi}_{\mathrm{R}}\,\hat{\mathbf{f}}_{\mathrm{R}}$$

which could be solved by an iterative fixed-point scheme.

The estimate $\hat{\mathbf{f}}_{\mathrm{R}}$ is thus a linear combination of the differences between training data and faulty measurement along the reconstruction directions $\boldsymbol{\Xi}_{\mathrm{R}}^{\top}$. The coefficients $\hat{\alpha}_j$, acting as training data contributions to the estimation, could be positive or negative but their sum is one, which may lead to compensation in the linear combination.

B. Estimation subject to nonnegativity and sum-to-one constraints

By analogy with [6], instead of directly estimating $\mathbf{f}_{\mathrm{R}}$, $\mathbf{f}_{\mathrm{R}}$ is defined as the weighted sum of the differences between the faulty observation and training data along the considered reconstruction directions, and the weight vector $\boldsymbol{\alpha}$. Therefore, the previous optimization problem (3) can be reformulated as the following constrained optimization problem:

$$\hat{\mathbf{f}}_{\mathrm{R}} = \arg\min_{\boldsymbol{\alpha}} SPE(\boldsymbol{z}_{\mathrm{R}})$$

with

$$\begin{cases} \boldsymbol{z}_{\mathrm{R}} &= \boldsymbol{x} - \boldsymbol{\Xi}_{\mathrm{R}}\,\mathbf{f}_{\mathrm{R}} \\ \mathbf{f}_{\mathrm{R}} &= \displaystyle\sum_{j=1}^{n}\alpha_j\,\boldsymbol{\Xi}_{\mathrm{R}}^{\top}\,(\boldsymbol{x} - \boldsymbol{x}_j) \end{cases} \tag{7}$$

subject to the constraints:

$$\alpha_j \geq 0, \qquad j = 1, \ldots, n$$
$$\sum_{j=1}^{n}\alpha_j = 1$$

To solve more easily this fully constrained nonlinear optimization problem, a procedure similar to that proposed by [8] is used:

- We introduce the following variable change:

$$\alpha_j = \frac{\omega_j}{\sum_{i=1}^{n}\omega_i} \quad \text{with} \quad \omega_j \geq 0,\, j = 1, \ldots, n \tag{8}$$

- We proceed to the minimization with respect to the new variable ω_j, subject to nonnegativity constraint only, using a component-wise gradient descent,
- We come back to the initial variables α_j.

The Lagrangian function for nonnegativity constraint problem is then given by:

$$\mathcal{L}(\boldsymbol{\omega}, \boldsymbol{\mu}) = SPE(\boldsymbol{z}_{\mathrm{R}}) - \boldsymbol{\mu}^{\top}\boldsymbol{\omega}$$

At the optimum $(\boldsymbol{\omega}^*, \boldsymbol{\mu}^*)$, the Karush-Kuhn-Tucker (KKT) conditions reduce to:

$$\omega_j^*\,\left.\frac{\partial SPE(\boldsymbol{z}_{\mathrm{R}})}{\partial \omega_j}\right|_{\omega_j = \omega_j^*} = 0 \qquad j = 1, \ldots, n \tag{9}$$

$$\left.\frac{\partial SPE(\boldsymbol{z}_{\mathrm{R}})}{\partial \omega_j}\right|_{\omega_j = \omega_j^*} \geq 0 \qquad j = 1, \ldots, n \tag{10}$$

$$\omega_j^* \geq 0 \qquad j = 1, \ldots, n \tag{11}$$

To solve this nonlinear equation (9) under the constraints (10) and (11), the following gradient descent updating scheme is defined:

$$\omega_j^{(t+1)} = \omega_j^{(t)} - \eta_j^{(t)}\,g_j(\omega_j^{(t)})\,\omega_j^{(t)}\,\frac{\partial SPE(\boldsymbol{z}_{\mathrm{R}}^{(t)})}{\partial \omega_j}$$

where $\eta_j^{(t)}$ is the step-size which controls the convergence of the algorithm, $g_j(\omega_j^{(t)}) > 0$ is a positive function scaling the gradient and $\frac{\partial SPE(\boldsymbol{z}_{\mathrm{R}}^{(t)})}{\partial \omega_j} = \left.\frac{\partial SPE(\boldsymbol{z}_{\mathrm{R}})}{\partial \omega_j}\right|_{\omega_j = \omega_j^{(t)}}$ to simplify notations.

This above equation could be re-written as follows:

$$\omega_j^{(t+1)} = \omega_j^{(t)}\,m_j^{(t)} \tag{12}$$

with

$$m_j^{(t)} = 1 - \eta_j^{(t)}\,g_j(\omega_j^{(t)})\,\frac{\partial SPE(\boldsymbol{z}_{\mathrm{R}}^{(t)})}{\partial \omega_j} \tag{13}$$

To avoid an expensive step-size computation for each component of $\boldsymbol{\omega}$, in practice, a single step-size $\eta^{(t)}$ is chosen.

Now let us come back to the initial variables (8). First, let us demonstrate the following equality:

$$\sum_{t=1}^{n} \omega_t^{(t+1)} = \sum_{t=1}^{n} \omega_t^{(t)}$$

Proof: By summing each side of the equality (12), we have $\sum_{j=1}^{n} \omega_j^{(t+1)} = \sum_{j=1}^{n} \omega_j^{(t)}$ if:

$$\sum_{j=1}^{n} \omega_j^{(t)}\, g_j(\omega_j^{(t)})\, \frac{\partial SPE(\boldsymbol{z}_{\mathrm{R}}^{(t)})}{\partial \omega_j} = 0 \tag{14}$$

First, let us evaluate the partial derivatives of $SPE(\cdot)$ with respect to ω_j:

$$\frac{\partial SPE(\boldsymbol{z}_{\mathrm{R}})}{\partial \omega_j} = \sum_{i=1}^{n} \frac{\partial SPE(\boldsymbol{z}_{\mathrm{R}})}{\partial \alpha_i}\, \frac{\partial \alpha_i}{\partial \omega_j}$$

with

$$\frac{\partial \alpha_i}{\partial \omega_j} = \frac{\delta_{ji} - \alpha_i}{\sum_{i=1}^{n} \omega_i}$$

Then the following equation is obtained:

$$\frac{\partial SPE(\boldsymbol{z}_{\mathrm{R}})}{\partial \omega_j} = \frac{1}{\sum_{i=1}^{n} \omega_i} \left(\frac{\partial SPE(\boldsymbol{z}_{\mathrm{R}})}{\partial \alpha_j} - \sum_{i=1}^{n} \alpha_i\, \frac{\partial SPE(\boldsymbol{z}_{\mathrm{R}})}{\partial \alpha_i} \right) \tag{15}$$

$\sum_{i=1}^{n} \omega_i^{(t)}$ being the same for all components, the scaling function is chosen as :

$$g_j(\omega_j^{(t)}) = \sum_{i=1}^{n} \omega_i^{(t)} \tag{16}$$

Let us replace (15) and (16) in (14) :

$$\begin{aligned} \sum_{j=1}^{n} \omega_j^{(t)}\, g_j(\omega_j^{(t)})\, \frac{\partial SPE(\boldsymbol{z}_{\mathrm{R}}^{(t)})}{\partial \omega_j} &= \sum_{j=1}^{n} \omega_j^{(t)}\, \frac{\partial SPE(\boldsymbol{z}_{\mathrm{R}}^{(t)})}{\partial \alpha_j} - \\ &\sum_{j=1}^{n} \frac{\omega_j^{(t)}}{\sum_{i=1}^{n} \omega_i^{(t)}} \sum_{i=1}^{n} \omega_i^{(t)}\, \frac{\partial SPE(\boldsymbol{z}_{\mathrm{R}}^{(t)})}{\partial \alpha_i} \\ &= 0 \end{aligned}$$

■

Since these two sums are equal, we divide respectively by $\sum_{i=1}^{n} \omega_i^{(t+1)}$ and $\sum_{i=1}^{n} \omega_i^{(t)}$ the left and right sides of (12) to obtain:

$$\alpha_j^{(t+1)} = \alpha_j^{(t)}\, m_j^{(t)} \tag{17}$$

Let us evaluate the partial derivatives of $SPE(\cdot)$ with respect to α_j:

$$\frac{\partial SPE(\boldsymbol{z}_{\mathrm{R}})}{\partial \alpha_j} = \frac{\partial SPE(\boldsymbol{z}_{\mathrm{R}})}{\partial \mathbf{f}_{\mathrm{R}}^{\top}}\, \frac{\partial \mathbf{f}_{\mathrm{R}}}{\partial \alpha_j} \tag{18}$$

From (7), we obtain:

$$\frac{\partial \mathbf{f}_{\mathrm{R}}}{\partial \alpha_j} = \boldsymbol{\Xi}_{\mathrm{R}}^{\top} (\boldsymbol{x} - \boldsymbol{x}_j) \tag{19}$$

Let us replace (18) and (19) in (15):

$$g_j(\omega_j^{(t)})\, \frac{\partial SPE(\boldsymbol{z}_{\mathrm{R}})}{\partial \omega_j} = (\boldsymbol{z}_{\mathrm{R}} - \boldsymbol{x}_j)^{\top}\, \boldsymbol{\Xi}_{\mathrm{R}}\, \frac{\partial SPE(\boldsymbol{z}_{\mathrm{R}})}{\partial f_{\mathrm{R}}}$$

Let us suppose that $\alpha_j^{(t)} > 0$, in order to ensure the nonnegativity of $\alpha_j^{(t+1)}$, the sign of $m_j^{(t)}$, defined in (13), has to be considered:

- If $\frac{\partial SPE(\boldsymbol{z}_{\mathrm{R}}^{(t)})}{\partial \omega_j} \leq 0$, then $m_j^{(t)} \geq 1$, the nonnegativity constraint is satisfied $\forall\, \eta^{(t)}$ and the weight $\alpha_j^{(t)}$ increases,
- if $\frac{\partial SPE(\boldsymbol{z}_{\mathrm{R}}^{(t)})}{\partial \omega_j} > 0$, then the step-size must be computed by a line search to ensure the convergence of the algorithm such as:

$$0 \leq \eta^{(t)} \leq \max_{j \in \mathrm{C}} g_j(\omega_j^{(t)})\, \frac{\partial SPE(\boldsymbol{z}_{\mathrm{R}}^{(t)})}{\partial \omega_j} \tag{20}$$
$$\text{with} \quad \mathrm{C} = \left\{ j \mid \frac{\partial SPE(\boldsymbol{z}_{\mathrm{R}}^{(t)})}{\partial \omega_j} > 0 \right\}$$

  then $0 \leq m_j^{(t)} \leq 1$ and the weight α_j decreases.

Let us also note that:

- since the step-size is chosen such that $m_j^{(t)}$ is positive (20) and taking into account that $\sum_{i=1}^{n} \alpha_i^{(t+1)} = \sum_{i=1}^{n} \alpha_i^{(t)}$, the vector $\alpha_j^{(t+1)}$ satisfies the nonnegativity and sum-to-one constraints as long as its initialization $\boldsymbol{\alpha}^{(0)}$ satisfies these two constraints,
- let us suppose that the maximal step in (20) is obtained for the i^{th} observation:

$$\eta_{\max}^{(t)} = \frac{1}{\left(\boldsymbol{z}_{\mathrm{R}}^{(t)} - \boldsymbol{x}_i\right)^{\top} \boldsymbol{\Xi}_{\mathrm{R}}\, \frac{\partial SPE(\boldsymbol{z}_{\mathrm{R}}^{(t)})}{\partial f_{\mathrm{R}}}}$$

  and that at iteration t, $\eta^{(t)} \to \eta_{\max}^{(t)}$ then:

$$m_j^{(t)} \to \left(1 - \frac{\left(\boldsymbol{z}_{\mathrm{R}}^{(t)} - \boldsymbol{x}_j\right)^{\top} \boldsymbol{\Xi}_{\mathrm{R}}\, \frac{\partial SPE(\boldsymbol{z}_{\mathrm{R}}^{(t)})}{\partial f_{\mathrm{R}}}}{\left(\boldsymbol{z}_{\mathrm{R}}^{(t)} - \boldsymbol{x}_i\right)^{\top} \boldsymbol{\Xi}_{\mathrm{R}}\, \frac{\partial SPE(\boldsymbol{z}_{\mathrm{R}}^{(t)})}{\partial f_{\mathrm{R}}}} \right)$$

  and the weight $\alpha_i^{(t)}$ tends to converge to zero. This could explain the sparse side-effect obtained with nonnegativity constraints.

Moreover, fixed points of the updating rule (17) occur when $\alpha_j = 0$ and $\frac{\partial SPE(\boldsymbol{z}_{\mathrm{R}})}{\partial \omega_j} > 0$ or when $\alpha_j > 0$ and $m_j = 1$, which implies that $\frac{\partial SPE(\boldsymbol{z}_{\mathrm{R}})}{\partial \omega_j} = 0$. These are consistent with the KKT conditions and thus they define the stopping rule of the iterative algorithm.

IV. Simulation on the CSTR benchmark

The efficiency of the proposed method is illustrated on a nonisothermal continuous chemical reactor (CSTR) benchmark defined in [9].

A. Model of the CSTR

We make certain assumptions : the reactor contains well mixed constituents and has constant physical properties. A first order reaction is taken into consideration where reactant A is mixed with a solvent in order to obtain a product B.

The mass balance and the energy balance describe the dynamic behavior of the CSTR as follows :

$$\begin{aligned}\frac{dC_A}{dt} &= \frac{F}{V}C_{A0} - \frac{F}{V}C_A - k_0 \mathrm{e}^{-\frac{E}{RT}} C_A \\ V\rho C_P \frac{dT}{dt} &= \rho C_P F(T_0 - T) + (-\Delta H_r) V k_0 \mathrm{e}^{-\frac{E}{RT}} C_A \\ &\quad - \frac{aF_C^{b+1}}{F_C + \frac{aF_C^b}{(2\rho_C C_{PC})}}(T - T_C)\end{aligned} \tag{21}$$

In the equation (21), V is the volume of the tank, F the input flow rate, ΔH_r the heat of reaction, k_0 the reaction velocity constant, ρ the reaction mixture density, ρ_C the coolant density, C_P the volumetric heat capacity, C_{PC} the coolant capacity, E the activation energy, R the gas constant, the temperature T, the cooling water temperature T_C, the inlet temperature T_0, the coolant flow F_C, and the outlet concentration C_A.

The overall flow rate of the CSTR is formed by the flow rate of the reactant and the solvent and the inlet concentration is determined by : $C_{A0} = \frac{C_{AA}\, F_A + C_{AS}\, F_S}{F_A + F_S}$, with the inlet concentrations C_{AA} and C_{AS}, the solvent flow F_S, and the reactant flow F_A. In the CSTR model, T is controlled by F_C using a proportional-integral (PI) controller.

Nine variables are measured for fault detection and isolation process defining an observation $\boldsymbol{x}$ at each instant with:

$$\boldsymbol{x} = [T_C \;\; T_0 \;\; C_{AA} \;\; C_{AS} \;\; F_S \;\; F_C \;\; C_A \;\; T \;\; F_A]^\top \tag{22}$$

A first-order autoregressive model defines all disturbances on the reactions, namely : $\boldsymbol{x}_t = \phi\, \boldsymbol{x}_{t-1} + \sigma_e\, \mathbf{e}_t$, where $\mathbf{e}_t$ is a Gaussian distribution and σ_e^2 is the variance of $\mathbf{e}_t$. In addition to these disturbances, Gaussian measurement noises with variance are added to the variables. More details can be found in [9].

The initial conditions are as follows : $T_0 = 370$ (K), $T_C = 365$ (K), F_C = 15 (m^3/min), $T = 368.25$ (K), F_S = 0.9 (m^3/min), F_A = 0.1 (m^3/min), C_A = 0.8 (kmole/m^3), C_{AS} = 0.1 (kmole/m^3), C_{AA} = 19.1 (kmole/m^3).
The parameters of the simulation are given by : V = 1 (m^3), $\rho = 10^6$ (g/ m^3), E/R= 8330.1 (K), C_P = 1 (cal/gK), C_{PC} = 1 (cal/gK), b = 0.5, $k_0 = 10^{10}$ (m^3/kmole $\times$ min), a = 1.678 $\times 10^6$ (cal/min K), $\Delta H_r = -1.3 \times 10^7$ (cal/kmole).
The parameters of the PI are : $K_C = -1.5$, $T_I = 5$.

A set of 100 observations were used during training, where KPCA is applied using a Gaussian kernel, defined by (1), with $c = 0.36$. 28 principal components were needed to define the KPCA model and a detection threshold was estimated with a value of 0.21.

In order to test the efficiency of the proposed method, two sets of 100 observations each were used with two different faults added as described in the following.

B. Fault affecting the inlet temperature T_0

The first fault affects the inlet temperature T_0. To this end, we add a step of $1.5K$ on the sensor when measuring the inlet temperature T_0 starting at instant 51, on the first set of 100 observations. As we can see from figure 1, the fault is detected and the SPE is only greater than the threshold 0.21 highlighted with red line, after instant 51.

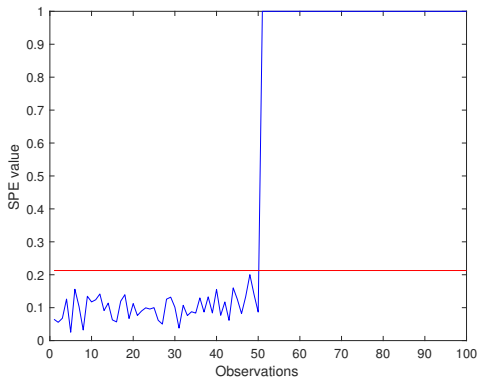

Fig. 1. Detection of fault on the inlet temperature T_0

Once, the fault was detected, we proceed with the reconstruction of each one of the nine variables of $\boldsymbol{x}$. Figure 2 illustrates the cumulative reconstruction-based contribution (RBC) defined by (4) evaluated on the 50 faulty observations. The cumulative RBC corresponds to the sum of RBC over the 50 faulty observations. As we can see, the second variable, that represents T_0 in observation $\boldsymbol{x}$ defined in (22), is the one with the highest contribution. We evaluate the SPE after reconstruction of T_0, the obtained value is 0.08 (compared to 1 before the reconstruction) which is less than the threshold estimated during the training process namely 0.21. Thus, the fault was corrected, T_0 was accurately reconstructed and the estimated fault amplitude is 1.49.

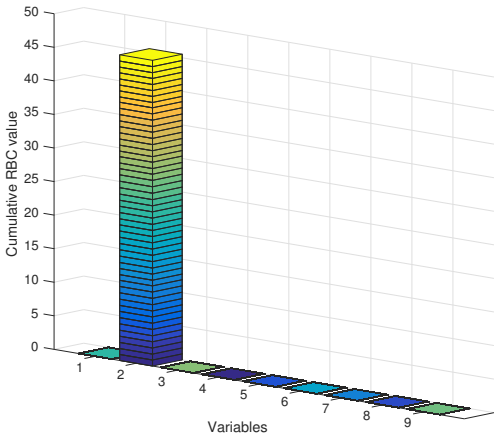

Fig. 2. Cumulative RBC for fault affecting T_0

C. Fault affecting the temperature T

Let us now introduce the second fault. To this end, a new set of 100 observations is used. A step of $1K$ is added on the temperature T starting at instant 51. The fault was correctly detected and a figure similar to figure 1 was obtained.

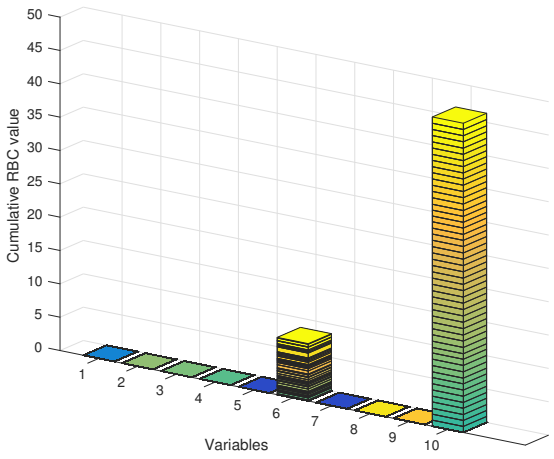

Fig. 3. Cumulative RBC for fault affecting T

Once, we detect the presence of fault, we seek the variable affected by the fault. Therefore, we apply the proposed method and reconstruct each one of the nine variables and evaluate the reconstruction-based contribution. Figure 3 shows the cumulative RBC of all faulty observations. As we can see, the first nine bars present the contribution when we reconstruct each one of the nine variables. Variable 6 which represents the coolant flow F_C has higher contribution by comparing it to the other eight variables. However, if we evaluate the SPE after the reconstruction of F_C, we obtain 0.82 which is greater than the threshold 0.21, indicating that the fault was not corrected.

It is worth nothing to mention that if we compare the contributions of F_C and T_0 in figure 2, we see that F_C has a smaller contribution making it easier to explain the fact that SPE is greater than the threshold. Therefore, we seek a combination of different variables for which the SPE after reconstruction is less than the threshold.

However, the fault on T has a complex effect on the system. This can be explained by the fact that T is controlled by a PI. The temperature, increasing due to sensor fault, will be corrected by increasing the coolant flow F_C. However, the real temperature inside the reactor is less than the one needed for normal functioning. In consequence, the concentration of the product at the output defined by C_A is increasing too. Then, the three variables T, F_C and C_A are affected by the fault on T. Thus, by reconstructing these three variables, the SPE obtained after reconstruction is 0.07 which is less than threshold. The contribution of these three variables is shown in the last bar (number 10) of figure 3. Clearly, the contribution of the combination of these three variables is greater than the contribution of reconstructing F_C alone and is similar to the contribution of T_0.

It is worth nothing to mention that, by comparing our results to those of Alcala and Qin [3], our proposed method shows the importance of reconstructing more than one variable using the contribution of figure 3. However, in [3], the authors evaluate an average RBC using the square of the estimated amplitude of fault, which is not similar to RBC evaluated as a difference between the SPE before and after reconstruction. The results in figure 4, in the aforementioned paper, indicate an average RBC for the variable F_C that is almost identical to the one after reconstructing the three variables. Thus, it is not clear why it is necessary to reconstruct more than one variable. Based on their results, it could have been enough to reconstruct the coolant flow F_C. This is not the case with our approach since the RBC between the two reconstructions are very different.

In such cases, where the fault has a complex effect on the system, the fault isolation will be subject to a structural-functional analysis of the system in order to determine the candidate subsets of variables to be reconstructed. Due to the fact that many variables were needed to correct the fault, it is important to study the propagation of the effect in order to get to the main cause. It is not the aim of this paper.

V. Conclusion

This paper proposes a sparse fault estimation method to evaluate the reconstruction-based contribution for multiple fault isolation. The fault magnitude estimation is here formulated as an optimization problem under nonnegativity and sum-to-one constraints. A multiplicative iterative scheme is proposed in order to solve it. The effectiveness of the proposed method is demonstrated on a simulated continuous stirred tank reactor (CSTR) process.

A future work is necessary to determine the useful potential reconstruction directions in order to handle the combinatorial nature of multiple fault isolation.

References

[1] J.-M. Lee, C. Yoo, S. W. Choi, P. A. Vanrolleghem, and I.-B. Lee, "Nonlinear process monitoring using kernel principal component analysis," *Chemical Engineering Science*, vol. 59, no. 1, pp. 223 – 234, 2004.

[2] S. W. Choi and I.-B. Lee, "Nonlinear dynamic process monitoring based on dynamic kernel pca," *Chemical Engineering Science*, vol. 59, no. 24, pp. 5897 – 5908, 2004.

[3] C. F. Alcala and S. J. Qin, "Reconstruction-based contribution for process monitoring with kernel principal component analysis," *Industrial & Engineering Chemistry Research*, vol. 49, no. 17, pp. 7849–7857, 2010.

[4] S. Mika, B. Schölkopf, A. Smola, K.-R. Müller, M. Scholz, and G. Rätsch, "Kernel pca and de-noising in feature spaces," in *Proceedings of the 1998 Conference on Advances in Neural Information Processing Systems II*. Cambridge, MA, USA: MIT Press, 1999, pp. 536–542.

[5] F. Sha, Y. Lin, L. K. Saul, and D. D. Lee, "Multiplicative updates for nonnegative quadratic programming," *Neural Computation*, vol. 19, no. 8, pp. 2004–2031, 2007.

[6] M. Kallas, P. Honeine, C. Richard, C. Francis, and H. Amoud, "Non-negativity constraints on the pre-image for pattern recognition with kernel machines," *Pattern Recognition*, vol. 46, no. 11, pp. 3066 – 3080, 2013.

[7] D. Bertsekas, *Nonlinear Programming*. Athena Scientific, 1999.

[8] H. Lantéri, C. Theys, and C. Richard, "Nonnegative matrix factorization with regularization and sparsity-enforcing terms," in *4th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, Dec 2011, pp. 97–100.

[9] S. Yoon and J. F. MacGregor, "Fault diagnosis with multivariate statistical models part i: using steady state fault signatures," *Journal of Process Control*, vol. 11, no. 4, pp. 387 – 400, 2001.