

Fault Detection and Diagnosis Using Support Vector Machines - A SVC and SVR Comparison

Davi L. de Souza¹, Matheus H. Granzotto², Gustavo M. de Almeida³, Luís C. Oliveira-Lopes^{2,*}

¹Dept. of Chemical Engineering, Federal University of the Triângulo Mineiro, Avenida Doutor Randolfo Borges Júnior, 1250, Univerdecidade, 38064-200 - Uberaba, MG - Brazil

²School of Chemical Engineering, Federal University of Uberlandia, Av. João Naves de Ávila, 2121, Bloco K, Santa Mônica, 38.400-902, Uberlândia, MG, Brazil

³Dept. of Chemical Engineering and Statistics, CAP, Federal University of Sao Joao del-Rey, Rod. MG 443, Km 07, Faz. do Cadete, 36.420-000, Ouro Branco, MG, Brazil

Abstract This paper presents the use of Support Vector Machines (SVM) methodology for fault detection and diagnosis. Two approaches are addressed: the SVM for classification (Support Vector Classification – SVC) and SVM for regression (Support Vector Regression – SVR). A comparison was made between the two techniques through the study of a reactor of cyclopentenol production. In the case studied, different fault scenarios were introduced and it was evaluated which technique was able to detect and diagnose them. Finally, a comparison was made between the fault detection methodologies based on SVM and Dynamic Principal Component Analysis (DPCA) based detection techniques for a jacketed CSTR.

Keywords Fault detection, Support vector machines, Process safety monitoring

1. Introduction

The monitoring of control systems is related to the ability of supervising the operation of industrial plants while evaluating the loss of performance caused by oscillations, disturbances, faults in sensors, and valve stiction. It also contains action such as diagnosing possible causes of problems that may degrade the productive capacity of the process, alarms management and providing strategies on how to act to maintain or even improve the operation efficiency.

Discovering abnormalities in control systems is a very important task. There are processes variations that might be connected to various sources, so, process plants containing control loops with poor performance are often found in an industrial scenario [1]. An important source of control degradation and safety issues are caused by faults in process control loops.

There are different techniques for fault detection in the literature [2-5]. Nowadays, the Support Vector Machine (SVM, also Support Vector Networks) is an alternative for fault detection and diagnostics. The original SVM algorithm was proposed by Vladimir N. Vapnik [6], and provides a powerful tool for pattern recognition [7-8] to deal with problems that have nonlinear, large and limited data sample.

The support vectors utilize a hyperplane with maximum margin to separate different classes of data producing a satisfactory overall performance. Thus, this methodology can provide a single solution with a strong regularized feature that is very suitable for classification problems poorly conditioned. The SVM technique has been used for various applications such as face recognition, time series forecasting [9], fault detection [10-11] and modeling of nonlinear dynamical systems [12]. This paper presents the results of fault detection in a reaction system for the production of cyclopentenol in a CSTR (Continuous Stirred Tank Reactor) with three simulated faults, utilizing the techniques of statistical machine learning support vector machine SVC and SVR, and for a jacketed CSTR with one simulated fault, the dimensionality reduction technique DPCA (Dynamic Principal Component Analysis) is also compared with the evaluated SVM techniques.

2. Methodologies for Fault Detection

2.1. Support Vector Machines for Classification (SVC)

In machine learning, support vector machines for classification (SVC) are supervised learning models with associated learning algorithms that analyze data and recognize patterns. The basic SVM takes a set of input data and predicts, for each given input, which of two possible classes form the output, making it a non-probabilistic binary linear classifier. Given a set of training examples, each marked as belonging to one of two categories, an SVC

* Corresponding author:

lcol@ufu.br (Luís C. Oliveira-Lopes)

Published online at <http://journal.sapub.org/safety>

Copyright © 2014 Scientific & Academic Publishing. All Rights Reserved

training algorithm builds a model that assigns new examples into one or other category. An SVC model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belonging to a category based on which side of the gap they are [13].

In addition to performing linear classification, SVCs can efficiently perform non-linear classification using what is called as kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.

More formally, a support vector machine constructs a hyperplane or set of hyperplanes in a high- or infinite-dimensional space, which can be used for classification, regression, or other tasks. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training data point of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier.

The idea of using SVC for separating two classes is to find support vectors (i.e. representative training data points) to define the bounding planes, in which the margin between the both planes is maximized. The number of support vectors increases with the complexity of the problem. To define SVC mathematically, the training data for the two classes are first stacked into an $n \times m$ matrix X , where n is the number of observations and m the number of variables.

Denote x_i as a column vector representing the i th row of X . An $n \times n$ diagonal matrix Y with +1 and -1 entries is then used to specify the membership of each x_i in class +1 or -1. In SVC, the prime problem is to separate the set of training vectors belonging to two separate classes,

$$D = \{(x^1, y^1), \dots, (x^l, y^l)\}, x \in \mathbb{R}^*, y \in \{-1, 1\} \quad (1)$$

with a hyperplane,

$$\langle w, x \rangle + b = 0 \quad (2)$$

The set of vectors is said to be optimally separated by the hyperplane if it is separated without error and the distance between the closest vectors to the hyperplane is maximal. There is some redundancy in Eq. 2, and without loss of generality it is appropriate to consider a canonical hyperplane [6], where the parameters w, b are constrained by

$$\min_i |\langle w, x^i \rangle + b| = 1 \quad (3)$$

This constraint on the parameterization is preferable to alternatives in simplifying the formulation of the problem. In words it states that: *the norm of the weight vector should be equal to the inverse of the distance, of the nearest point in the data set to the hyperplane.*

The distance $d(w, b; x)$ of a point x from the hyperplane (w, b) is according to

$$d(w, b; x) = \frac{|\langle w, x \rangle + b|}{\|w\|} \quad (4)$$

The optimal hyperplane is given by maximizing the margin ρ , subject to the constraints of Eq. 3. The margin is given by

$$\begin{aligned} \rho(w, b) &= \min_{x^l: y^l = -1} d(w, b; x^l) + \min_{x^l: y^l = 1} d(w, b; x^l) \\ &= \min_{x^l: y^l = -1} \frac{|\langle w, x^l \rangle + b|}{\|w\|} + \min_{x^l: y^l = 1} \frac{|\langle w, x^l \rangle + b|}{\|w\|} \\ &= \frac{1}{\|w\|} \left(\min_{x^l: y^l = -1} |\langle w, x^l \rangle + b| + \min_{x^l: y^l = 1} |\langle w, x^l \rangle + b| \right) \\ &= \frac{2}{\|w\|} \end{aligned} \quad (5)$$

Hence the hyperplane that optimally separates the data is the one that minimizes

$$\Phi(w) = \frac{1}{2} \|w\|^2 \quad (6)$$

It is independent of b provided Eq. 3 is satisfied (i.e. it is a separating hyperplane) changing b will move it in the normal direction to itself. Accordingly the margin remains unchanged but the hyperplane is no longer optimal in that and it will be nearer to one class than the other. To consider how minimizing Eq. 6 is equivalent to implementing the SRM principle, suppose that the following bound holds in

$$\|w\| < A \quad (7)$$

Then from Eq. 3 and Eq. 4,

$$d(w, b; x) \geq \frac{1}{A} \quad (8)$$

The SVC has to be trained with data from normal operations and faulty conditions of the system, making it possible to detect the type of failure. The system builds a vector with all the classified failures for all available data.

2.2. Support Vector Machines for Regression (SVR)

The SVM for regression (SVR) utilizes the normal operating data to build a model that predicts outputs for determined inputs.

The SVR foresees the results for every input applied to the model, resulting in a difference between the real value and the predicted value for the output variables.

SVMs can also be applied to regression problems by the introduction of an alternative loss function [14]. The loss function must be modified to include a distance measure.

Similarly to the classification problem, a non-linear model is usually required to adequately model plant data. In the same manner as the non-linear SVC approach, a non-linear mapping can be used to map the available plant data into a high dimensional feature space where linear regression is performed. The kernel approach is again employed to address the dimensionality. The non-linear SVR solution, using an ϵ -insensitive loss function, is given by

$$\begin{aligned} \max_{\alpha, \alpha^*} W(\alpha, \alpha^*) &= \max_{\alpha, \alpha^*} \sum_{i=1}^l \alpha_i^* (y_i - \epsilon) - \alpha_i (y_i - \epsilon) \\ &\quad - \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l (\alpha_i^* - \alpha_i) (\alpha_j^* - \alpha_j) K(x_i, x_j) \end{aligned} \quad (9)$$

with constraints,

$$\begin{aligned} 0 &\leq \alpha_i, \alpha_i^* \leq C, \quad i = 1, \dots, l \\ \sum_{i=1}^l (\alpha_i - \alpha_i^*) &= 0 \end{aligned} \quad (10)$$

Solving Eq. 9 with constraints Eq. 10 to evaluate the Lagrange multipliers, α_i , α_i^* , and the regression function is given by

$$f(x) = \sum_{SVs} (\bar{\alpha}_i - \bar{\alpha}_i^*) K(x_i, x) + \bar{b} \quad (11)$$

where

$$\begin{aligned} \langle \bar{w}, x \rangle &= \sum_{i=1}^l (\bar{\alpha}_i - \bar{\alpha}_i^*) K(x_i, x_j) \\ \bar{b} &= -\frac{1}{2} \sum_{i=1}^l (\bar{\alpha}_i - \bar{\alpha}_i^*) (K(x_i, x_r) + K(x_i, x_s)) \end{aligned} \quad (12)$$

As with the SVC the equality constraint may be dropped if the Kernel contains a bias term, b , being accommodated within the Kernel function and the regression function is given by

$$f(x) = \sum_{i=1}^l (\bar{\alpha}_i - \bar{\alpha}_i^*) K(x_i, x) \quad (13)$$

The optimization criteria for the other loss functions are similarly obtained by replacing the dot product with a kernel function. The ϵ -insensitive loss function is attractive because unlike the quadratic and Huber cost functions, where all the plant data will be support vectors, the SV solution can be sparse. The quadratic loss function produces a solution which is equivalent to ridge regression, or zeroth order regularization, where the regularization parameter is given by

$$\lambda = \frac{1}{2c} \quad (14)$$

The fault detection happens when divergence between the predicted output data and the actual real output data takes place. If the divergence is larger than a certain number, in this case used as 3σ (three times the standard deviation of training normal operation data), the fault is detected.

The system builds a vector with all the instants where the fault was detected or not. The SVR is not capable of identifying the type of fault occurred, because this methodology utilizes only the data points of normal operation condition of the plant.

2.3. Dynamic Principal Component Analysis (DPCA)

The PCA technique is used to build statistical models based on historical data of the process, indicated primarily for large industrial processes, with lots of important variables for process control.

With the statistical model obtained by PCA, it is possible to detect failures using the most important variables of the process, designing the data even in a reduced dimensional space, *i.e.* all the process information is preserved, however, the PCA technique allows using a data set of reduced size and which captures the system variability.

Several researchers [15-19] have used the PCA as a tool for monitoring industrial processes, because this technique allows reducing the size of a data set of a multivariable process being analyzed and has a simple implementation [20].

Consider the matrix of historical data $X \in \mathbb{R}^{n \times m}$ containing n samples of m process variables collected under normal operation. This matrix must be normalized to zero mean and unitary variance with the scale parameter vectors

\bar{x} and σ as the mean and variance vectors, respectively. The next step to calculate PCA is to construct the covariance matrix S :

$$S = \frac{1}{n-1} X^T X = V \Lambda V^T \quad (15)$$

with the diagonal matrix $\Lambda = \Sigma^T \Sigma \in \mathbb{R}^{n \times m}$ contains the eigenvalues λ of real non-negative and decreasing magnitude ($\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_m \geq 0$).

The main objective of PCA is to capture the variations of the data while minimizing the effect of the possible presence of random noise, since they affect the PCA representation, so it is very common to use the value a (number of principal components) highest eigenvalues λ to ensure the main objective of the technique. This dimension reduction is motivated to protect the approach from detecting systems failure that is in fact random noise [21].

With the a highest eigenvalue belonging to the columns of the matrix V it is possible to write the matrix $P \in \mathbb{R}^{m \times a}$, so:

$$T = X P \quad (16)$$

The matrix T contains the projection of the observations in X in a smaller space, and the projection of T , in the m -dimensional observation space is:

$$\hat{X} = T P^T \quad (17)$$

The residual matrix E can be determined by the difference of X and \hat{X} :

$$E = X - \hat{X} \quad (18)$$

Finally the original data space can be calculated by:

$$X = T P^T + E \quad (19)$$

a) Number of components (a) to be retained in a PCA model

In literature, there are various techniques for obtaining the number of principal components. These techniques are intended to decouple the changes in state of the random variations to this, determining the appropriate number of eigenvalues that must be maintained in the model PCA. The most common techniques are:

- Scree procedure;
- Cumulative percent variance (CPV), which can be obtained according to:

$$CPV(a) = \frac{\sum_{i=1}^a \lambda_i}{\text{trace}(S)} \quad (20)$$

- Prediction residual sum of squares;
 - Cross-validation procedure;
 - Parallel analysis: it has the highest performance when compared with other techniques and is frequently used [20].
- An algorithm for the calculation proposed in [21] as follows:

1. generate a set of data normally distributed with zero mean and unitary variance with the same dimension as the real data set (m variables and n observations);
2. do a PCA on the data;
3. get the eigenvalues sorted in decreasing order;
4. plot the eigenvalues of the original data along with data normally distributed;
5. get a through the intersection between the profiles.

So far, what has been discussed using the PCA technique

for monitoring control systems, does not take into account the statistical dependence on past observations, *i.e.*, the technique only considers observations in a given time, which in industrial processes that statement is not valid due to the small time for sampling, which in many cases are in the order of seconds [21]. The statistical independence is achieved only for sampling intervals from 2-12 h [22].

One way to account for the effect of this dependence for processes with short time of sampling intervals is to take into account the temporal correlations, doing now with the PCA method is extended with the previous observations g in each observation vector, as follows [21]:

$$X(k - g, :) = [X_k^T \ X_{k-1}^T \ X_{k-2}^T \ \dots \ X_{k-g}^T] \quad (21)$$

$$k = 1, 2, \dots, n$$

with x_k^T the observation vector of dimension m in the sampling instant k .

This method is known as dynamic PCA or DPCAm [21]. Studies were performed to obtain automatically g [23], however, experiments indicate that $g = 1$ or 2 is acceptable, when using PCA in process monitoring.

b) Fault Detection

The most common techniques used in the detection and diagnosis of faults in multivariable processes are: Hotelling T^2 Statistics and Q Statistics (square prediction error - SPE). These techniques were applied in this work, which is aimed at detecting possible faults in control loops. One can calculate the statistic T^2 as follows [22]:

$$T^2 = x^T P \Lambda_\alpha^{-1} P^T x \quad (22)$$

where Λ_α is a squared matrix formed by the first a rows and columns of Λ by PCA model, and the process is considered *normal* for a given significance level α if

$$T^2 \leq T_\alpha^2 = \frac{(n^2-1)a}{n(n-a)} F_\alpha(a, n-a) \quad (23)$$

where $F_\alpha(a, n-a)$ is the critic value of the *Fisher-Snedecor* distribution with α the level of significance that takes values between 90% and 95%.

The Q statistic can be calculated by:

$$Q = r^T r$$

$$r = (I - PP^T)x \quad (24)$$

The limits of this statistic can be calculated by:

$$Q_\alpha = \theta_1 \left[\frac{h_0 c_\alpha \sqrt{2\theta_2}}{\theta_1} + 1 + \frac{\theta_2 h_0 (h_0 - 1)_2}{\theta_1^2} \right]^{\frac{1}{h_0}} \quad (25)$$

where c_α is the value of the normal distribution with α as the level of significance.

3. Case Studies

3.1. Case study #1 - Cyclopentenol Reactor

A cyclopentenol reactor is investigated for three different

fault patterns and the results for fault detection and diagnosis indicate that the SVC and SVR used have greater reliability and faster detection. The SVM methods used for faults diagnostics seem to deliver better results for the scenarios investigated compared to the dimensionality reduction method.

Consider a reaction mechanism known as *van der Vusse* reaction [24]. The major reaction is the transformation of cyclopentadiene (component A) in the product cyclopentenol (component B). A parallel reaction occurs producing a byproduct, dicyclopentadiene (component D). Furthermore, cyclopentenol reacts again forming an unwanted product cyclopentanediol (component C). All of these reactions can be described by the following reaction scheme



The reactor inlet contains only low reactant A in concentration C_{A0} . Assuming that the density of the liquid is constant and a distribution of an ideal residence time inside the reactor, the reactor dynamics equations *van der Vusse* (Figure 1) [25]. The reaction coefficient rates k_1 , k_2 and k_3 depend exponentially on the reactor temperature according to the Arrhenius law.

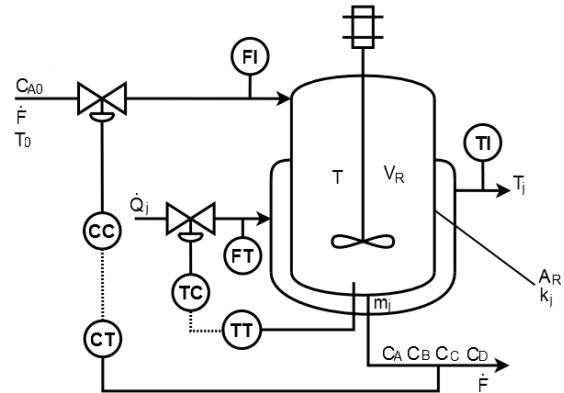


Figure 1. Reactor for producing cyclopentenol [25]

It is assumed that the reactor temperature, the concentration of cyclopentenol in the reactor, the temperature of the cooling jacket, the flow rate of heat removed and the reagent are obtained by measuring instruments. For the purposes of simulation, we added a Gaussian noise of mean 0 and variance 1×10^{-5} for concentration and 1×10^{-3} for other measurements. The input flow of reactant A, \dot{F} , and the amount of heat removed by the refrigerant, \dot{Q}_j , are the manipulated variables and subject to the following restrictions presented in

$$\begin{array}{l} 50 \frac{L}{h} \leq \dot{F} = u_1 \leq 350 \frac{L}{h} \\ -8500 \frac{kJ}{h} \leq \dot{Q}_j = u_2 \leq 0 \frac{kJ}{h} \end{array} \quad (27)$$

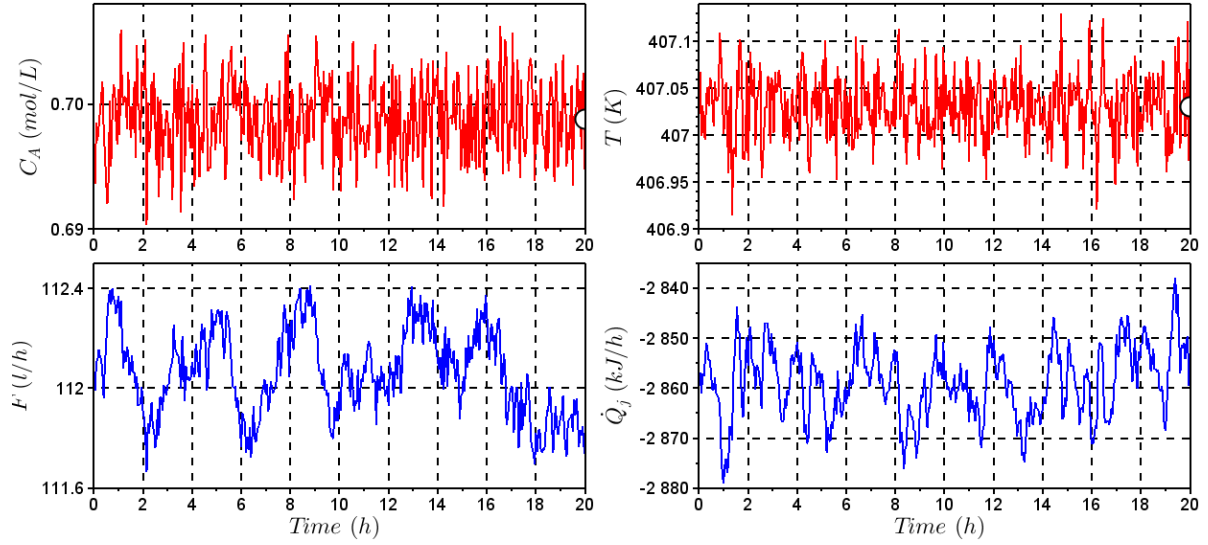


Figure 2. Behavior of the input and output variables – Normal operation

Table 1. Symbols and nominal values with units

h	Level in reactor (0.6 m)
q_F	Feed flow rate of reactor feed stream (1.67 L/s)
q	Flow rate of reactor outlet stream (1.67 L/s)
q_C	Coolant flow rate (0.25L/s)
A	Cross-sectional area of reactor (0.167 m ²)
T	Temperature in reactor (402.35 K)
T_F	Temperature of reactor feed stream (320 K)
ΔH_r	Heat of reaction (– 50 kJ/mol)
C_p	Heat capacity of reactor contents (kJ/(kg K))
C_{pC}	Heat capacity of coolant (kJ/(kg K))
C_A	Concentration of species A in reactor (0.0372 mol/L)
C_{AF}	Concentration of species A in reactor feed stream (1 mol/L)
k_0	Reaction pre-exponential factor (1.2×10^9 s ⁻¹)
E_a	Reaction activation energy (kJ/mol)
R	Universal gas constant (kJ/(mol K))
E_a/R	8750 K
U	Heat transfer coefficient (kJ/(s K m ²))
A_C	Area available for heat transfer (m ²)
$U A_C$	0.834 kJ/(s K)
T_C	Temperature of coolant in cooling jacket (345.44 K)
T_{CF}	Temperature of coolant feed (300 K)
ρ	Density of reactor contents (kg/L)
ρC_p	0.239 kJ/(L K)
ρ_C	Density of coolant (kg/L)
$\rho_C C_{pC}$	4.175 kJ/(L K)
V_C	Volume of the jacket (10 L)

It was shown that for a constant rate of heat removed and a variation 50 – 1500 l/h feed of reactant to the reactor, this process exist for six regions with different degrees of non-linearity [26]. The normal operation simulated for this case study is presented in Figure 2.

3.2. Case study #2 – A non-isothermal CSTR

The process used for the case study is a non-isothermal CSTR [27]. This case was studied because of the wide range of fault types and conditions available. A schematic diagram of the non-isothermal CSTR model is shown in Figure 3.

The nonlinear mass and energy balances are given by the following equation

$$\begin{aligned}
 \frac{dh}{dt} &= \frac{q_F - q}{A} \\
 \frac{dC_A}{dt} &= \frac{1}{Ah} (q_F C_{AF} - q C_A) - k_0 C_A e^{\frac{-E_a}{RT}} \\
 \frac{dT}{dt} &= \frac{1}{Ah} (q_F T_F - q T) + \frac{U A_C}{\rho C_p A h} (T_C - T) + \\
 &\quad + \frac{(-\Delta H_r)}{\rho C_p} k_0 C_A e^{\frac{-E_a}{RT}} \\
 \frac{dT_C}{dt} &= \frac{q_C}{V} (T_{CF} - T_C) - \frac{U A_C}{\rho_C C_{pC} V_C} (T_C - T) \quad (28)
 \end{aligned}$$

The level (h) and temperature (T) PI control, as seen in Figure 3, are tuned in appropriate dimensions as $K_C = -3$, $\tau_I = 90$ s, and $K_C = -0.2$, $\tau_I = 18$ s, manipulating the variables q and q_C , respectively.

To illustrate the application of the methods presented in this study, it was created the following faulty scenario: It was considered a sensor failure of CSTR level after 1200s, caused by instrument damage, causing an incorrect measurement of 3% less than last correct measurement. Figures 4 and 5 show the behavior of the control system before the fault taking place. It was noted that a sensor failure caused instability in the control loop because of the incorrect

information of the sensor. It was not possible for the manipulated variables to operate in another region to compensate for sensor failure.

Table 1 represents the symbols and units for the non-isothermal CSTR.

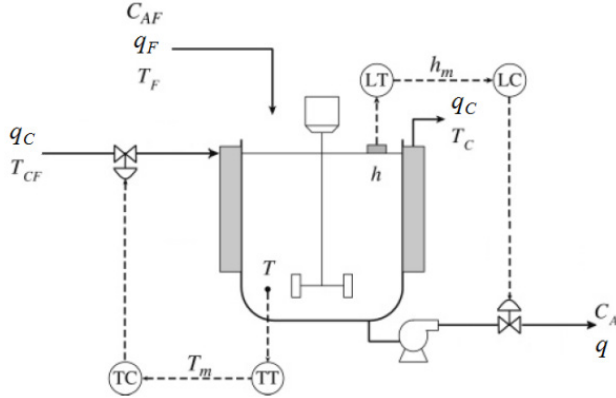


Figure 3. Non isothermal CSTR

4. Results and Discussion

4.1. Case study #1 - Cyclopentenol Reactor

For this system, it was chosen a flow constraint from 50 to 350 l/h. To control the process were designed two PID (Proportional-Integral-Derivative) controllers, one for controlling the concentration of product output and another for controlling the reactor temperature. The setpoint value for the concentration of cyclopentenol and reactor temperature were 0.69883 mol/l and 407.031 K, respectively. These values are relative to a steady state operation, with a reactant feed flow rate of 112 l/h (u_1) and a rate of removed heat of -2856.91 kJ/h (u_2).

The process and subsequent detection and diagnosis of faults in the production process of cyclopentenol were performed through computer simulation with the free mathematical software SciLab®. For illustration, it is considered the existence of two faulty scenarios in the process operation [25].

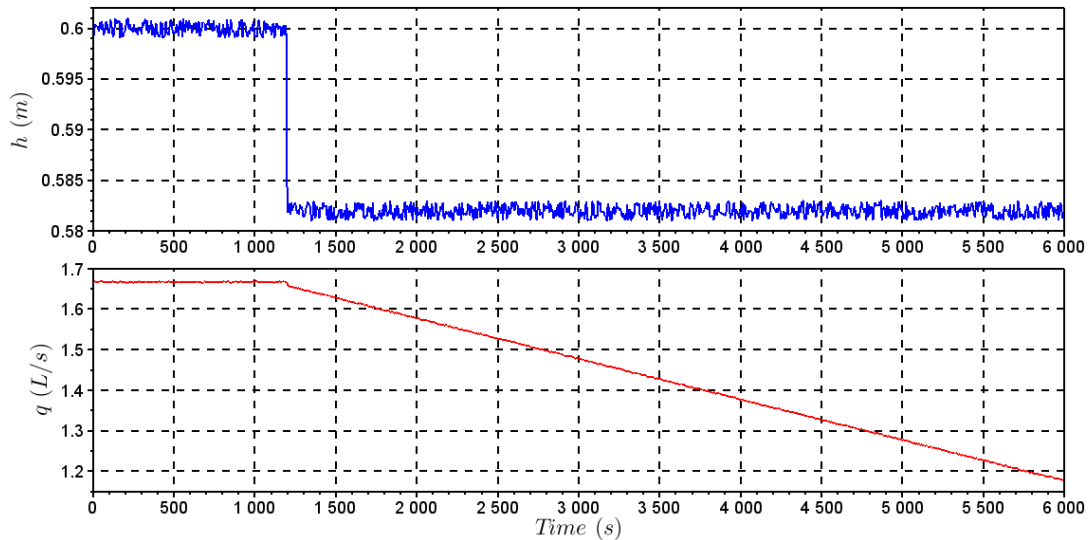


Figure 4. Behavior of h and q with the fault in the level sensor

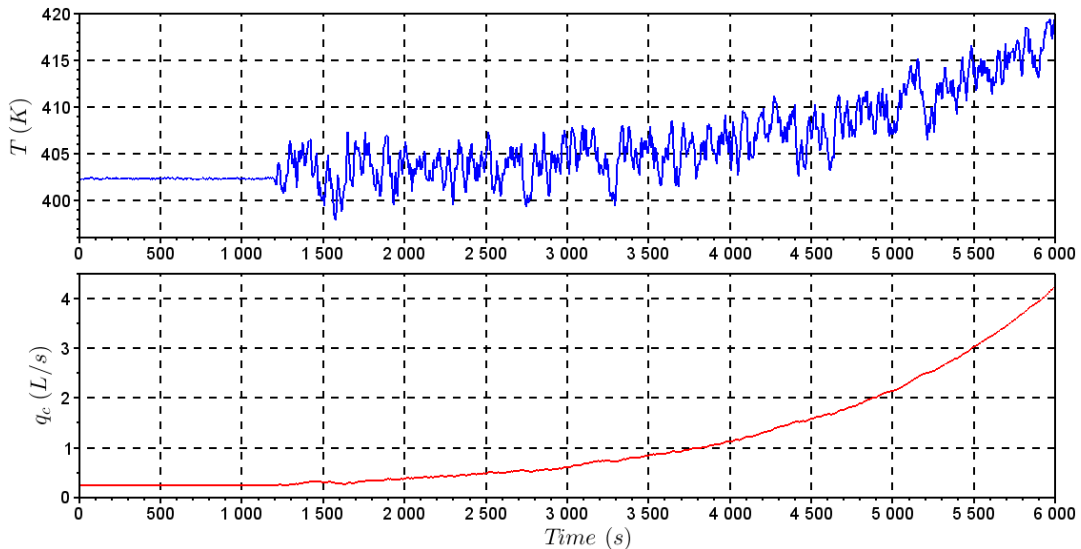


Figure 5. Behavior of T and q_c with the fault in the level sensor

The fault #01 considers that the reactor temperature sensor is gets damaged in a certain instant giving a 1% higher value than the last correct measurement produced by the sensor. To the reactor temperature measurement was added a random noise generated by a normal distribution with zero mean and 1×10^{-3} variance. Figure 5 shows the behavior of the output and input variables with a fault #01 taking place at the time instant of 8h.

The fault #02 was simulated by making a blocking in the valve of reactant flow to give a flow 30% lower than the one at steady state. Figure 6 shows output and input variables with a fault #02 taking place at the time instant of 8h.

The results for the operating conditions investigated for the CSTR are summarized in Table 3. The results contain performance metrics for fault detection and diagnosis with SVC and SVR. The algorithms SVC and SVR were applied using the LibSVM [28] library in SciLab[®]. The parameters for SVC were chosen as $C = 137.187$ and core (kernel) given by a radial basis $K(u, v) = \exp(-\gamma|u - v|^2)$ function where $\gamma = 1910.852$. These parameters were

found through a search, aiming at the best model for the SVC. For SVR, the parameters are $C = 100$ and core (kernel) given by the radial basis function with $\gamma = 0.5$ and $\epsilon = 0.05$.

The methods used were able to find the instant in which the fault was recognized (TFR) and the moment when the fault has been diagnosed correctly for the first time (TF).

To assess the quality of the fault detection methodology, the delay detection (TAD) in hours, which is the amount of time elapsed since the instant at which the fault took place and was correctly diagnosed for the first time was evaluated, and the indices of Eqs. 29 - 32 were introduced.

$$D_{F/F} = \frac{(\text{\# of sampling with faulty scenario detected})}{(\text{\# of sampling with faulty operation})} \quad (29)$$

$$D_{F/N} = \frac{(\text{\# of sampling with faulty scenario detected})}{(\text{\# of sampling with normal operation})} \quad (30)$$

$$D_{N/F} = \frac{(\text{\# of sampling with normal operation detected})}{(\text{\# of sampling with faulty operation})} \quad (31)$$

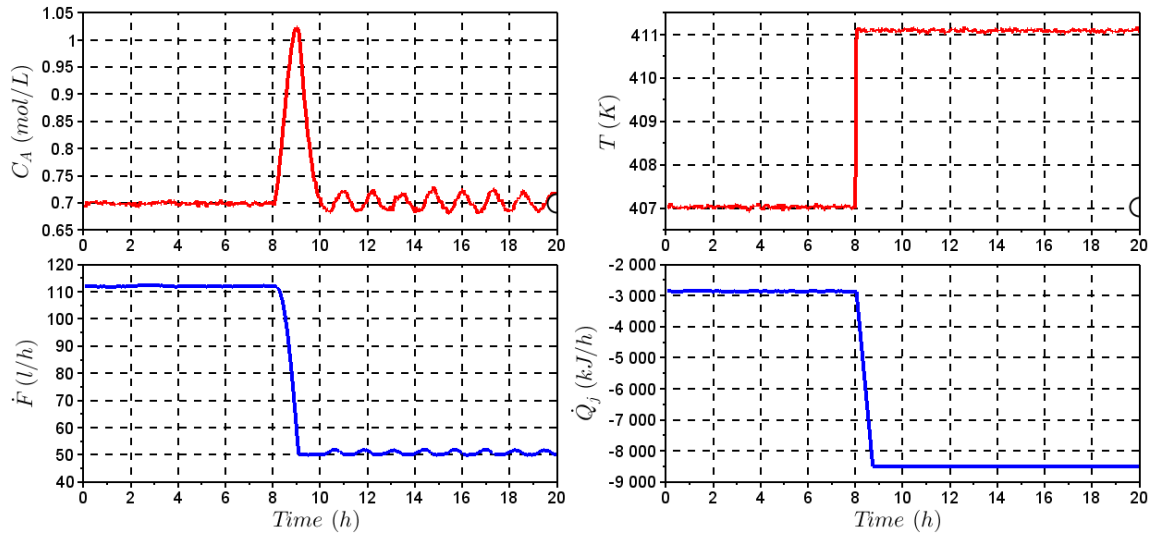


Figure 6. Behavior of the input and output variables – Fault #01

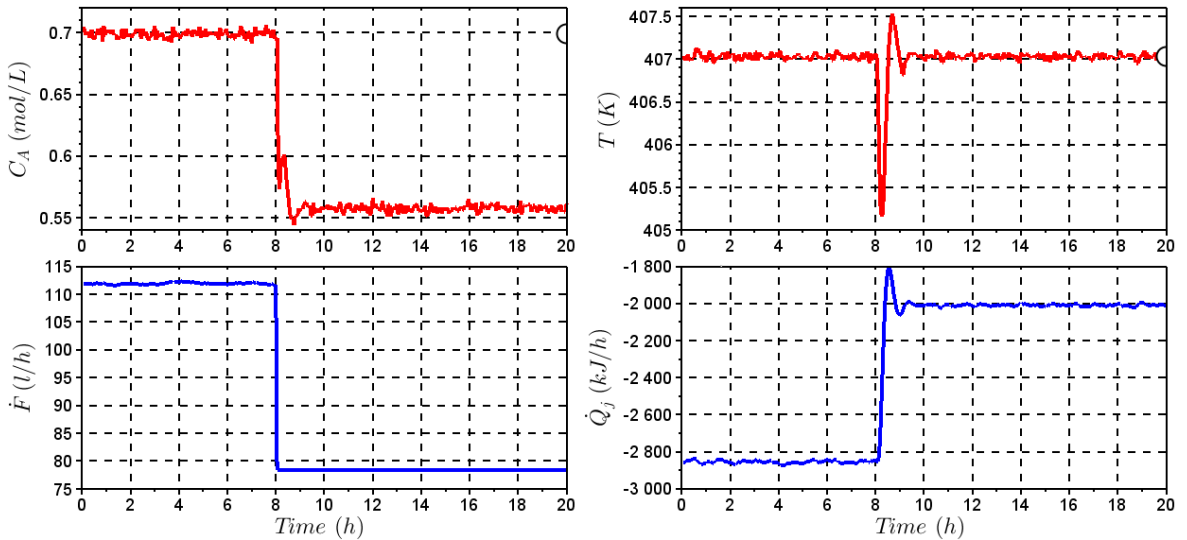


Figure 7. Behavior of the input and output variables – Fault #02

$$D_{N/N} = \frac{(\text{\# of sampling with normal operation detected})}{(\text{\# of sampling with normal operation})} \quad (32)$$

Table 3. Simulation Results for Case study #1 (20h of Operation, 0.05h Sampling Time. Fault Taking Place at Instant 8h)

Oper.	Fault #01		Fault #02	
Method	SVC	SVR	SVC	SVR
TFR (h)	8.55	8.10	8.10	8.15
TF (h)	8.50	8.05	8.05	8.10
$D_{F/F}$	100.0%	100.0%	100.0%	100.0%
$D_{F/N}$	4.33%	0.41%	0.41%	1.66%
$D_{N/F}$	0.00%	0.00%	0.00%	0.00%
$D_{N/N}$	95.67%	99.59%	99.59%	98.34%
TAD (h)	0.50	0.05	0.05	0.10

4.2. Case study #2 - Non-isothermal Reactor

4.2.1. Principal Component Analysis (PCA)

The technique DPCAm (Dynamic Principal Component Analysis) was used in this study for detecting a failure in the level sensor. We used parallel analysis technique to

determine the number of dimensions being removed from the PCA model, in this example, which has six measured variables (h, T, C_A, T_C, q and q_C) a was found to be equal to 3, $a = 3$. The cumulative percent variance (CPV) was 95.82%.

Given by Eq. 20 the data matrix is built with two delays ($g = 2$), and this technique DPCAm causes the X dimension to increase, for example, for normal operating data there are 6 measured variables with the delay of three sampling time, and 1001 observations (1001 samples) for each variable, making the dimension of $X \in \mathbb{R}^{1798 \times 27}$.

Figure 8 shows the statistical T^2 and Q applied to the “experimental” data collected. Note that the statistics are below the limits specified for the indication of failure, calculated by Eq. 23 and Eq. 24, respectively. It also set an alarm region, with a limit of 10% higher than calculated by Eq. 23 and Eq. 24. Figure 9 shows the T^2 and Q statistics for the level sensor failure. It was noted that at the moment when the failure has been simulated, (after 1200s) the methods are instantly able to indicate the presence of failure, since the statistics T^2 and Q were well above the limits calculated by Eq. 23 and Eq. 24, showing the efficiency of the technique.

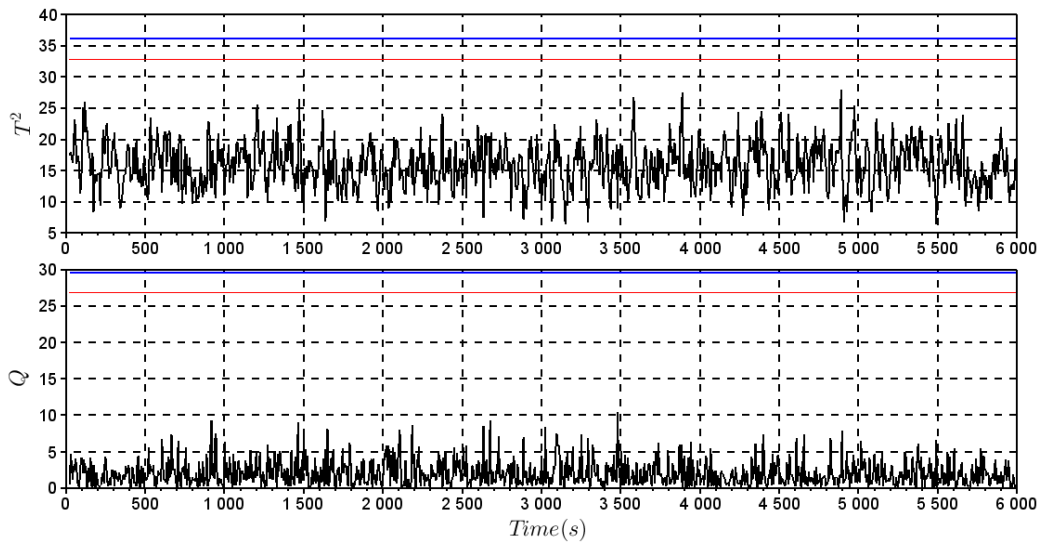


Figure 8. T^2 and Q statistics for data without fault

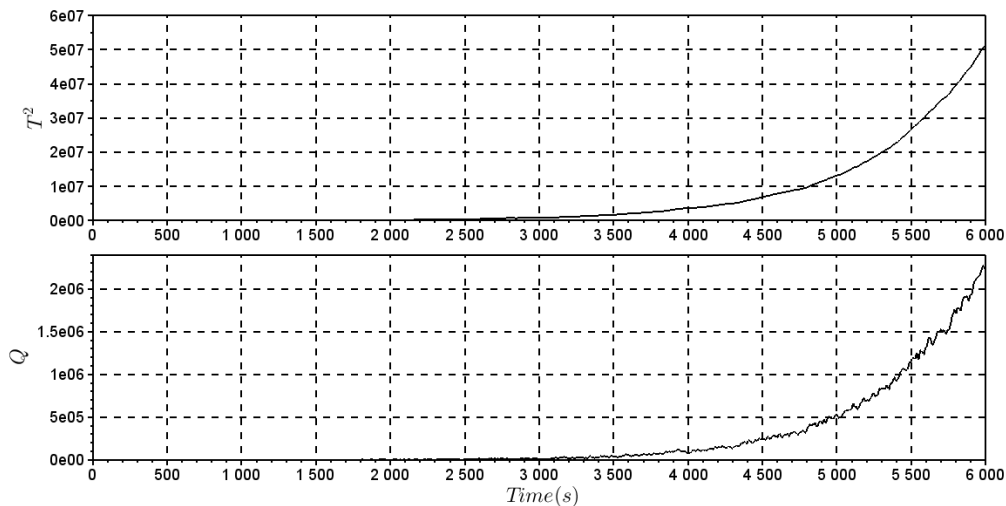


Figure 9. T^2 and Q statistics for data with fault

4.2.2. Support Vector Machine (SVM)

a) SVM for Classification (SVC)

When the SVM for classification was trained, the normal operation and fault(s) data points are utilized. The LibSVM was used for building the model, and it returns an accuracy of 99.8% for the model. It spent three sampling times for the model to detect the failure, applied in the time of 1200s. Figures 10 and Figure 11 show the behavior of the control system utilized as an example for fault detection with SVC. Figure 12 shows the representation for the classification over the time, where 0 is for normal operation and 1 for faulty operation.

b) SVM for Regression (SVR)

When the fault detection algorithm for SVM regression is applied, only the data for normal operation are used for

training the model. Once the model predicts output data of the system, it is important to compare the actual output of the system with the output provided from the model. When the actual data and predicted move away from each other, hence it is configured a system failure.

For this case, it is utilized a kernel with a radial bases function $K(x) = \exp(-\gamma|u - v| + C)^d$, where $\gamma = 0.5$, $C = 100$ and $d = 3$.

Figures 10 and 11 show the behavior of the control system with the fault utilized for fault detection with the SVM for regression. It took three sampling times for this methodology to detect the failure, applied at the time of 1200s. Figure 13(a) shows the representation for the classification over the time, where 0 is for normal operation and 1 for faulty operation. Figure 13(b) shows the predicted data and the real data for the simulation.

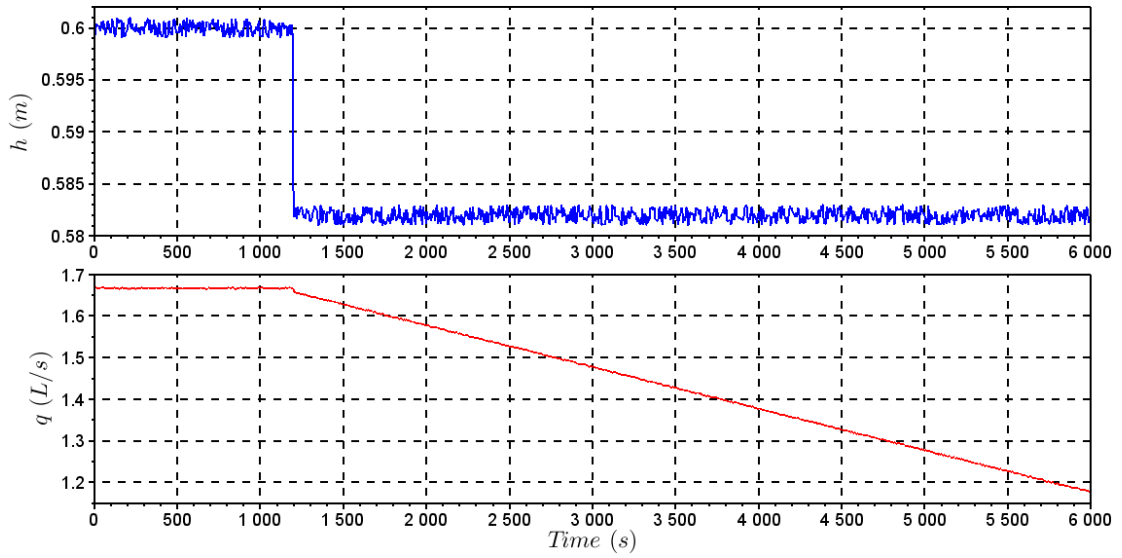


Figure 10. Behavior of h and q with the fault in level sensor

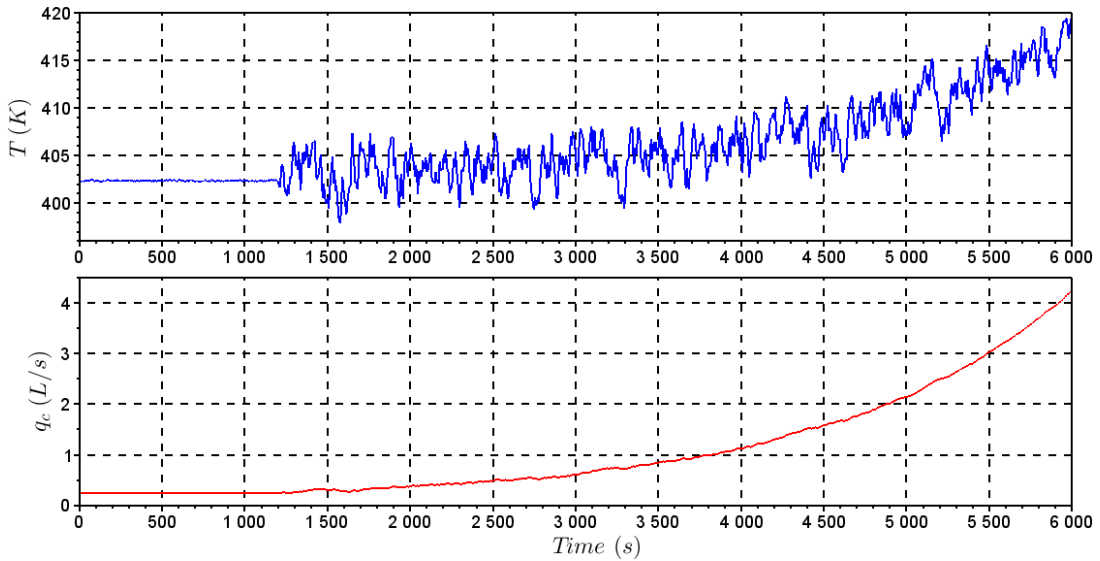


Figure 11. Behavior of T and q_c with the fault in level sensor

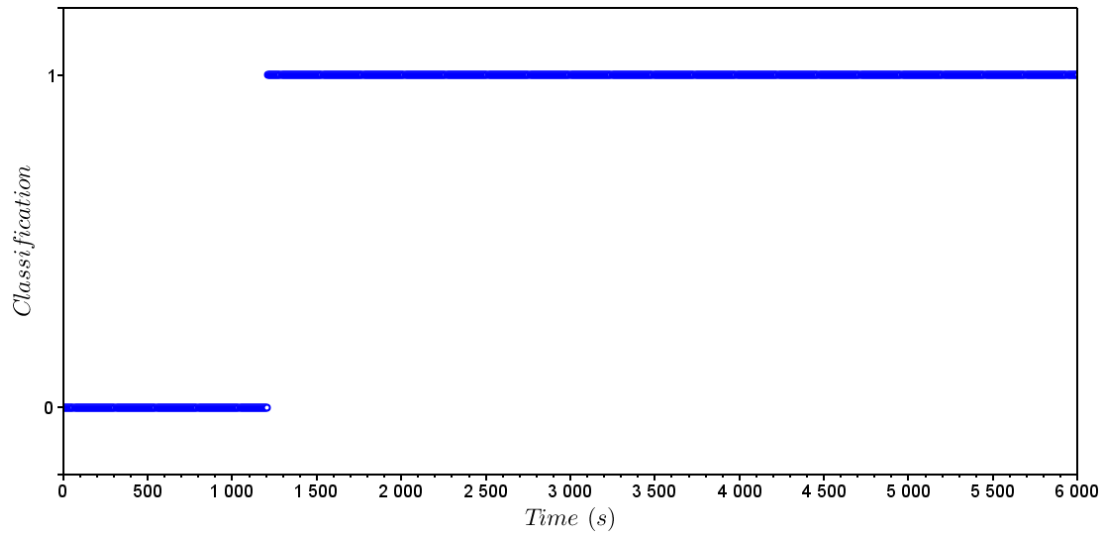


Figure 12. SVC - Detection. 0 – Normal operation; 1 – Faulty Operation

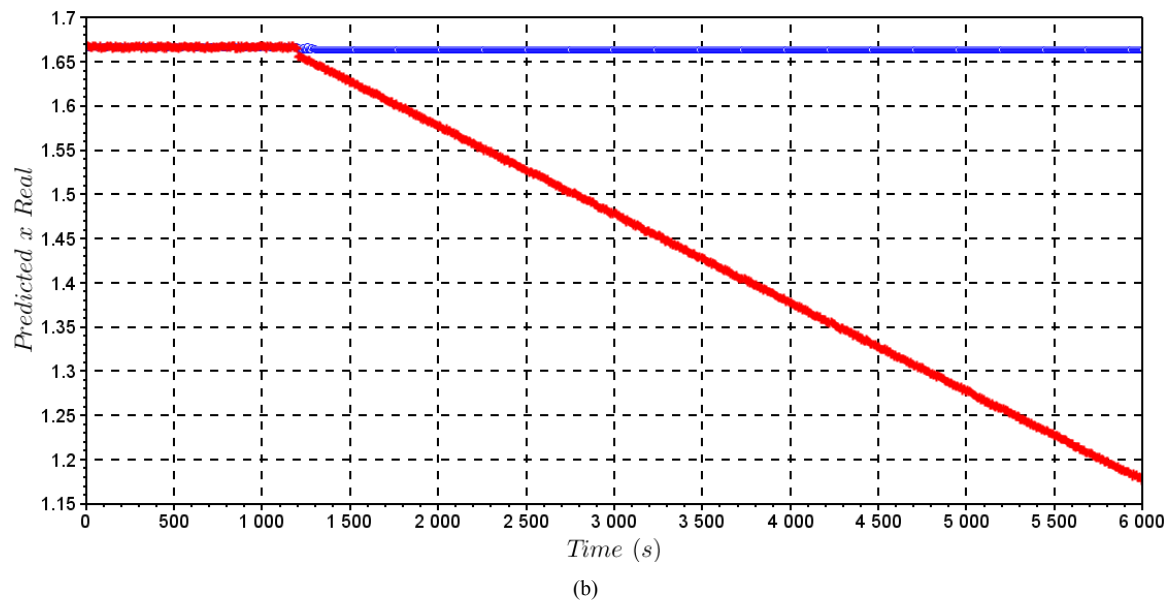
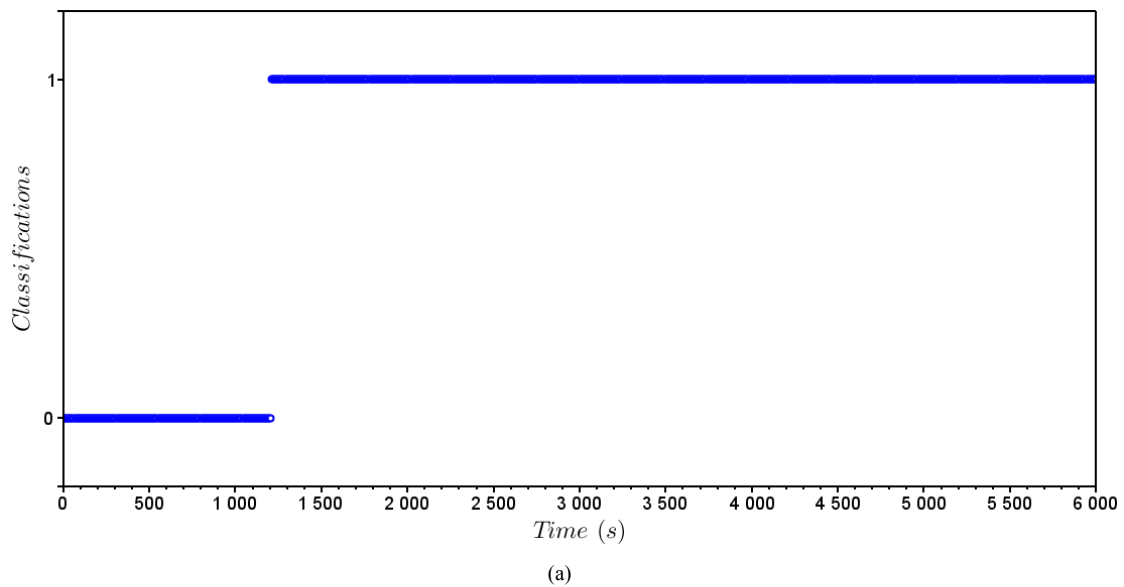


Figure 13. (a) SVM for Regression - Detection. 0 – Normal operation; 1 – Faulty Operation; (b) Predicted Data vs. Real Data

5. Conclusions

The SVC and SVR are new methods for detection and diagnosis of failures. The SVM methodology is promising for process monitoring in situations where process efficiency and industrial safety are addressed by an automatic monitoring system. The results for the cyclopentenol reactor with two failures show that although both methodologies may be used for detecting faults, it seems that SVR is faster than the SVC to detect failures, but these results might depend on the specific problem. Overall, both methods gave satisfactory results. Nevertheless, SVC has one great advantage over SVR, it has the ability of diagnosing faults. To conclude, both methodologies could be used simultaneously in a process monitoring system, taking advantage of the fast detecting time of the SVR approach and the classification capability of the SVC based methodology.

The SVM methods compared with the PCA show that the SVM methodology, with less information than the PCA, is better than the classic method for fault detection to the non-isothermal reactor for the faulty scenario evaluated.

ACKNOWLEDGEMENTS

The authors thank FAPEMIG, CAPES and CNPq Brazilian Research Foundations for the financial support.

REFERENCES

- [1] Y. Yamashita. An Automatic Method for Detection of Valve Stiction in Process Control Loops. *Control Eng. Practice*, 2006, 503–510.
- [2] H. P. Huang; M. C. Wu. Monitoring and Fault Detection for Dynamic Systems using Dynamic PCA on Filtered Data. In *Proc. Syst. Eng.*; Chen, B. and Westerberg, A. W., Eds.; Elsevier Science B.V.: Amsterdam, 2003.
- [3] D. Zhimin; J. Xinqiao; W. Lizhou. Fault Detection and Diagnosis based on improved PCA with JAA Method in VAV Systems. *Build. Environ.*, 2007, 42, 3221–3232.
- [4] X. B. Yang; X.Q. Jin; Z.M. Du; Y.H. Zhu. A Novel Model-Based Fault Detection Method for Temperature Sensor using Fractal Correlation Dimension. *Build. Environ.*, 2011, 46, 970–979.
- [5] M. J. Piovoso; K. Kosanovich; R. K. Pearson, Monitoring Process Performance in Real Time. *Proceedings of The American Control Conference*, Piscataway, New Jersey, Jun 24-26, 1992.
- [6] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, N.Y. ISBN 0-387-94559-8, 1995.
- [7] C. J. C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition, *Data Mining and Knowledge Discovery* 2, 121-167, 1998.
- [8] J. Lu; K. N. Plataniotis; A. N. Venetsanopoulos. Face recognition using feature optimization and mu-support vector learning, neural networks for signal processing XI, in *Proceedings of the IEEE signal, processing society workshop*. 373-382, 2001.
- [9] J.J Ahn.; K.J Oh; T.Y. Kim; D.H Kim. Usefulness of support vector machine to develop an early warning system for financial crisis, *Expert Systems with Applications*, Volume 38, Issue 4, 2966-2973, 2011.
- [10] K.C. Gryllias; I.A. Antoniadis. A Support Vector Machine approach based on physical model training for rolling element bearing fault detection in industrial environments, *Engineering Applications of Artificial Intelligence*, Volume 25, Issue 2, 326-344, 2012.
- [11] J. Park; I.-H. Kwon; S.-S Kim; J.-G Baek. Spline regression based feature extraction for semiconductor process fault detection using support vector machine, *Expert Systems with Applications*, Volume 38, Issue 5, 5711-5718, 2011.
- [12] Q. Wu. Car assembly line fault diagnosis model based on triangular fuzzy Gaussian wavelet kernel support vector classifier machine and genetic algorithm, *Expert Systems with Applications*, Volume 38, Issue 12, 14812-14818, 2011.
- [13] C. Cortes; V. Vapnik, Support-Vector Networks, *AT&T Labs-Research*, USA, 1995.
- [14] A. J. Smola. Regression estimation with support vector learning machines. Master's thesis, Technische Universität München, 1996.
- [15] M. J. Piovoso; K. Kosanovich, Applications of Multivariate Statistical Methods to Process Monitoring and Controller Design. *Int. J. Control*, 1994, 59, 743–765.
- [16] B. Wise; N. Gallagher. The Process Chemometrics Approach to Process Monitoring and Fault Detection. *J. Proc. Control*, 1996, 6, 329–348.
- [17] M. Misra; H. H. Yue; S. J. Qin; C. Ling. Multivariate Process Monitoring and Fault Diagnosis by Multi-Scale PCA. *Computers and Chemical Engineering*, 2002, 26, 1281–1293.
- [18] S. Dinga; P. Zhanga; E. Dingb, S. Yina; A. Naika; P. Dengc; W. Guic. On the Application of PCA Technique to Fault Diagnosis. *Tsinghua Sci. Technol.*, 2010, 15, 138–144.
- [19] A. Alkaya; I. Eker. Variance Sensitive Adaptive Threshold-Based PCA Method for Fault Detection with Experimental Application. *ISA Trans.*, 2011, 50, 287–302.
- [20] D. L. Souza. Metodologia para o Monitoramento de Sistemas de Controle na Indústria Química. Ph.D. Thesis, Federal University of Uberlândia, November 2011.
- [21] L. L. G. Reis. Controle Tolerante com Reconfiguração Estrutural Acoplado a Sistemas de Diagnóstico de Falhas. Dissertation. (M.Sc.), Federal University of Uberlândia, November 2008.
- [22] L. H. Chiang; E. L. Russel; R. D. Braatz. *Fault Detection and Diagnosis in Industrial Systems*. Springer: London, 2001.
- [23] W. Ku; R. Storer; C. Georgakis. Disturbance Detection and Isolation by Dynamic Principal Component Analysis. *Chemometr. Intell. Lab. Syst.*, 1995, 30, 1, 179–196.
- [24] K. U. Klatt, S. Engell. Gain-scheduling trajectory control of

- a continuous stirred tank reactor. *Computers & Chemical Engineering*, 491–502, 1998.
- [25] L.L.G. Reis; L. C. Oliveira-Lopes. Controle PID Tolerante a Falhas de um CSTR. In: *V Seminário Nacional de Controle e Automação Industrial, Elétrica e de Telecomunicações*, Salvador. V SNCA, 1-7, 2007.
- [26] B. A. Ogunnaike, W. H. Ray Process Dynamics, Modeling, and Control. *Oxford University Press*, New York, 1994.
- [27] J. S. Conner; D. E. Seborg Assessing the Need for Process Re-Identification. *Ind. Eng. Chem. Res.*, 2005, 44, 2767–2775.
- [28] C.-C. Chang; C.-J. Lin. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2011, 2:27:1--27:27.