Title

A graphical user interface for PCA-based MSPC: a benchmark software for multivariate statistical process control in MATLAB

Author names and affiliations

Pedro Villalba[a], Javier Sanchis[b], Alberto Ferrer[a]

[a]Departamento de Estadística e Investigación Operativa Aplicadas y Calidad, Universitat Politècnica de València, Camino de Vera, s/n, 46022 Valencia, Spain

[b]Instituto Universitario de Automática e Informática Industrial, Universitat Politècnica de València, Camino de Vera, s/n, 46022 Valencia, Spain

Corresponding author

Tel.: +34 96 3877493; fax: +34 96 3877499.

E-mail: aferrer@eio.upv.es (A. Ferrer)

Abstract

A Graphical User Interface (GUI) is developed in MATLAB as a tutorial for understanding the PCA-based MSPC strategy. The software allows users to analyze both simulated and external data sets. Simulated data are obtained from a nonlinear model of a binary distillation column implemented in Simulink. The nonlinear model has four manipulated variables, four controlled variables and three input measured disturbances, plus 41 molar fractions corresponding to every column stage. The methodology for PCA-based MSPC is implemented in two phases. During Phase I, the user can simulate the distillation column under normal operating conditions at three different operating points. When the simulation is finished, the GUI obtains the corresponding PCA model automatically. In Phase II, the user can simulate several scenarios with different combinations of disturbances and failures and monitor them using Squared Prediction Error (SPE) and T^2 control charts. Contribution plots are used to diagnose the original variables responsible of such abnormal situations. The software also incorporates the possibility to analyze external multivariate process datasets.

Keywords

Multivariate Statistical Process Control; Principal Component Analysis; latent variable; multivariate control charts; contribution plots; benchmark; nonlinear distillation column; tutorial; GUI

1. Introduction

Statistical Process Control (SPC) concepts and methods have become very important in the manufacturing and process industries [(1)]. Their objective is to monitor the performance of a process over time to verify that the process is remaining in a "state of statistical control". Such a state of control is said to exist if certain process or product variables remain close to their desired values and the only source of variation is "common-cause" variation, that is, variation that affects the process all the time and is essentially unavoidable given the particularities of the current process. SPC charts such as Shewhart, *CUSUM* and *EWMA* charts are used to monitor key product variables in order to detect the occurrence of any event having a "special" or "assignable" cause. By finding assignable causes, long-term improvements in the process and in product quality can be achieved by eliminating (or implementing) the causes improving the process or its operating procedures.

It is important to note that both the concepts and methods of SPC are complementary to those of automatic feedback process control. In general the two approaches are totally complementary. Automatic feedback control should be applied wherever possible to reduce variability in important process and product variables. Feedback controllers compensate for the predictable component of disturbances in important variables by adjusting other process variables and thereby transferring the variability into these less important manipulated variables. SPC monitoring methods should be applied on top of the process and its automatic control system in order to detect process behavior that indicates the occurrence of a special event. By diagnosing causes for the special events and removing them (rather than simply continuing to compensate for them), the process is improved.

Conventional Multivariate Statistical Process Control (MSPC) schemes are focused on monitoring the stability of the process mean. They are based on developing control charts from the Hotelling's T^2 statistic based on the original *K* registered (usually product quality or dimensional) variables [(2)]

$$T^2 = (\mathbf{z} - \boldsymbol{\mu})^T \mathbf{S}^{-1} (\mathbf{z} - \boldsymbol{\mu}) \quad (1)$$

where $\mathbf{z}$ is a (*K* x 1) vector of measurements; $\boldsymbol{\mu}$ is the in-control (*K* x 1) mean vector; and $\mathbf{S}$ is an estimate of the in-control (*K* x *K*) covariance matrix $\boldsymbol{\Sigma}$. This approach assumes that $\mathbf{z} \sim N_K(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and checks if the mean vector of the process $\boldsymbol{\mu}$ remains stable (assuming a constant covariance matrix). Once the multivariate control chart signals an out-of-control alarm it is needed to diagnose an assignable cause for it. This involves two steps: first (diagnostic) find which measured variable(s) contributes to the out-of-control signal, and second (corrective) determine what happens in the process that upsets the behavior of these variables.

Although conventional MSPC is well sounded from a statistical point of view, it suffers from lack of applicability in data-rich environments, typical of modern processes. This serious drawback comes from the fact that, as shown in Eq. (1), Hotelling's T^2 statistic in the original data space needs the inversion of the estimated covariance matrix **S**. To avoid problems with this inversion, the number of multivariate observations or samples (*N*) has to be larger than the number of variables (*K*), and covariance matrix **S** has to be well-conditioned (slightly correlated variables). Add to it, complete data (no missing values) are required to work out the Hotelling's T^2 statistic for any particular sample. Nevertheless, these requirements are not met in highly automated processes.

Latent variable methodology exploits the correlation structure of the original variables by revealing the few independent underlying events (latent variables) that are driving the process at any time. Multivariate statistical projection methods such as principal component analysis [(3)] (PCA) are used to reduce the dimensionality of the monitoring space by projecting the information in the original variables down onto low-dimensional subspaces defined by a few latent variables. The process is then monitored in these latent subspaces by using a few multivariate control charts built from multivariate statistics which can be thought of as process performance indices, or process wellness indices [(4)]. These charts retain all the simplicity of presentation and interpretation of conventional single variable SPC charts. However, by using the information contained in all the measured variables simultaneously, they are much more powerful for detecting out-of-control conditions. Another advantage of this methodology is that missing and noisy data are easily handled. If both process variables and product quality data are available, multivariate statistical predictive models based on projection to latent structures like PLS [(5)] (Partial Least Squares) can also be used.

This paper is divided into several sections and a conclusions summary. Section 2 introduces the distillation column used for the benchmark. Section 3 explains the GUI developed in MATLAB to implement the PCA-based MSPC strategy to the distillation column. Section 4 illustrates the results obtained from data simulated from the Simulink-based model of a binary distillation column and also from external datasets. Finally, conclusions are summarized in Section 5.

2. Column distillation benchmark

Consider a distillation column that has four controlled variables and four manipulated variables (Fig. 1). The controlled variables are product compositions, y_D (distillate composition) and x_B (bottom composition) and the liquid holdups in the reflux drum and reboiler, M_D and M_B, respectively [(6)]:

$$\mathbf{y} = (y_D\ x_B\ M_D\ M_B)^T \quad (2)$$

The four manipulated variables are product flow rates at the top (*D*) and at the bottom (*B*), and internal flow rates at the top (*L*) and at the bottom (*V*) of the column:

$\mathbf{u} = (L\ V\ D\ B)^T$ (3)

The feed stream is assumed to come from an upstream unit. Thus, the feed flow rate *F* cannot be manipulated, but it can be measured and used for feed forward control. Other disturbances are temperature (T_F) and composition of the feed (z_F).
In almost all industrial control configurations, the distillation column is first stabilized by closing two decentralized (SISO) loops for levels, involving the following

$\mathbf{y_2} = (M_D\ M_B)^T$; $\mathbf{u_2} = (D\ B)^T$ (4)

The two SISO loops for controlling y_2 are based on proportional controllers and are the following:

1. Distillate holdup level (M_D) controlled by distillate flow (*D*).
2. Bottom holdup level (M_B) controlled by bottom flow (*B*).

The remaining outputs are then the product compositions. In this paper, the LV-configuration is implemented (Fig. 1), which uses internal flows L and V to control these compositions

$\mathbf{y_1} = (y_D\ x_B)^T$; $\mathbf{u_1} = (L\ V)^T$ (5)

The two SISO loops for controlling $\mathbf{y_1}$ are based on proportional-integral controllers and are the following:

1. Distillate composition (y_D) is controlled by top internal flow (*L*).
2. Bottom composition (x_B) is controlled by bottom internal flow (*V*).

The LV-configuration is good from the point of view that the effect of $\mathbf{u_1}$ on $\mathbf{y_1}$ is nearly independent of the tuning of the level controllers (involving $\mathbf{y_2}$ and $\mathbf{u_2}$) (6).

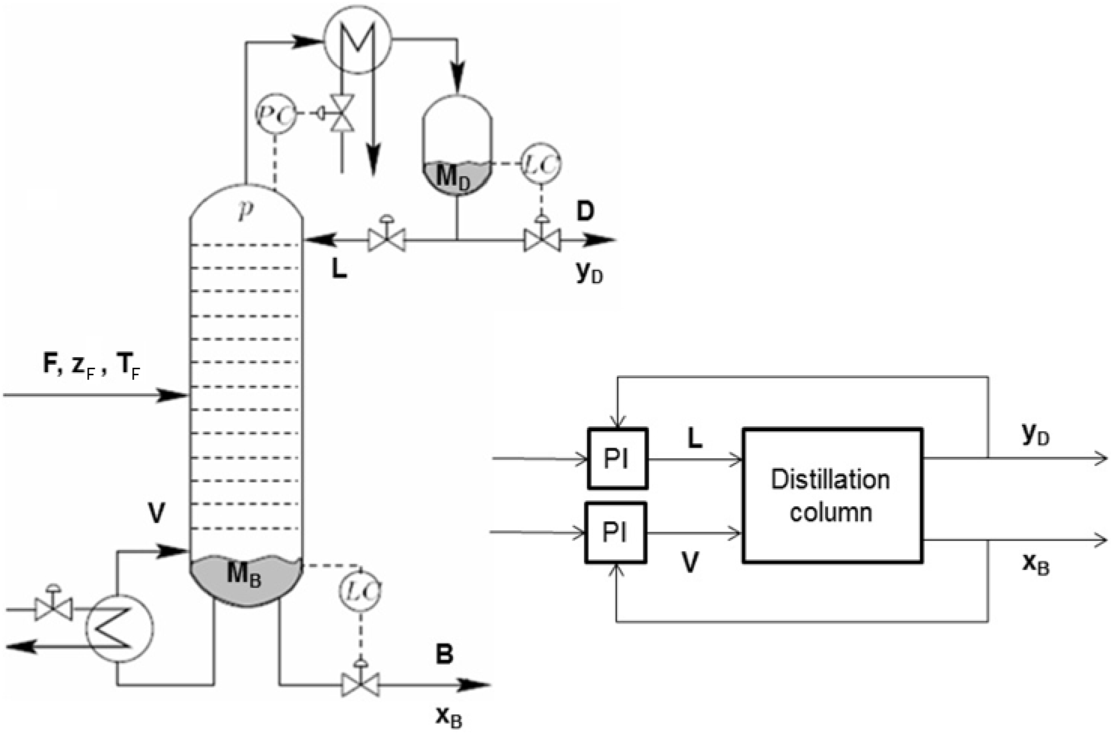


Fig. 1.- Distillation column with *L*-*V* configuration.

The model used in this work corresponds to a binary distillation column with the above control configuration and is based on a Simulink model[1] created by Skogestad [(7)]. This nonlinear model has four manipulated inputs (*L*, *V*, *D* and *B*), three disturbances (*F*, z_F and *q*) and the compositions and holdups for every stage of the column. The column has N_S=41 stages: the reboiler, the condenser and the 39 trays inside the column.

The model is applied to a methanol-ethanol mixture. Several enhancements are done to obtain an industrial-like model: noise is added to measurements, molar flows are converted into volumetric flows and original quality factor *q* is computed through feed composition z_F and feed temperature T_F (see Fig.S-1).

3. PCA-based MSPC software

A Graphical User Interface (GUI) is developed in MATLAB to implement the PCA-based MSPC strategy to the distillation column shown before[2]. This has been successfully tested in different MATLAB versions (2009–2017) (Mathworks, Sheborn, MA).

The PCA-based MSPC monitoring scheme, as any SPC scheme, is carried out in two phases. In Phase I (model building) monitoring charts are built according to a set of historical in-control data, once the performance of the process has been understood and modeled, and the

[1] The Simulink model is available at:
http://www.nt.ntnu.no/users/skoge/book/matlab_m/cola/cola.html

[2] Available at http://www.mathworks.es/matlabcentral/fileexchange/47169-a-benchmark-software-for-multivariate-statistical-process-control and developed by GIEM (Grupo de Ingeniería Estadística Multivariante, http://mseg.webs.upv.es/index.html

assumptions of its behavior and process stability are checked. In Phase II (model exploitation) these charts are used to monitor the process using on-line data, assuming the form of the distribution to be known along with its values of the in-control parameters [(8)].

Fig. 2 shows the main window with the toolbar and application menus that summarize all the functionalities. The menu is divided into three main groups. The first one ("File") takes into account options related to file management, the second one ("Phase I") deals with options used to develop a PCA model during Phase I, and the last one ("Phase II") is used to generate several test to simulate and monitor failures during Phase II.

The application allows starting a new benchmark from scratch or opening a previously saved one. User can save its progress at any time as well as export simulation results to an Excel file. Each sheet in this Excel file corresponds to a test in the benchmark. If Excel is not present, the software will attempt to write file in CSV format.

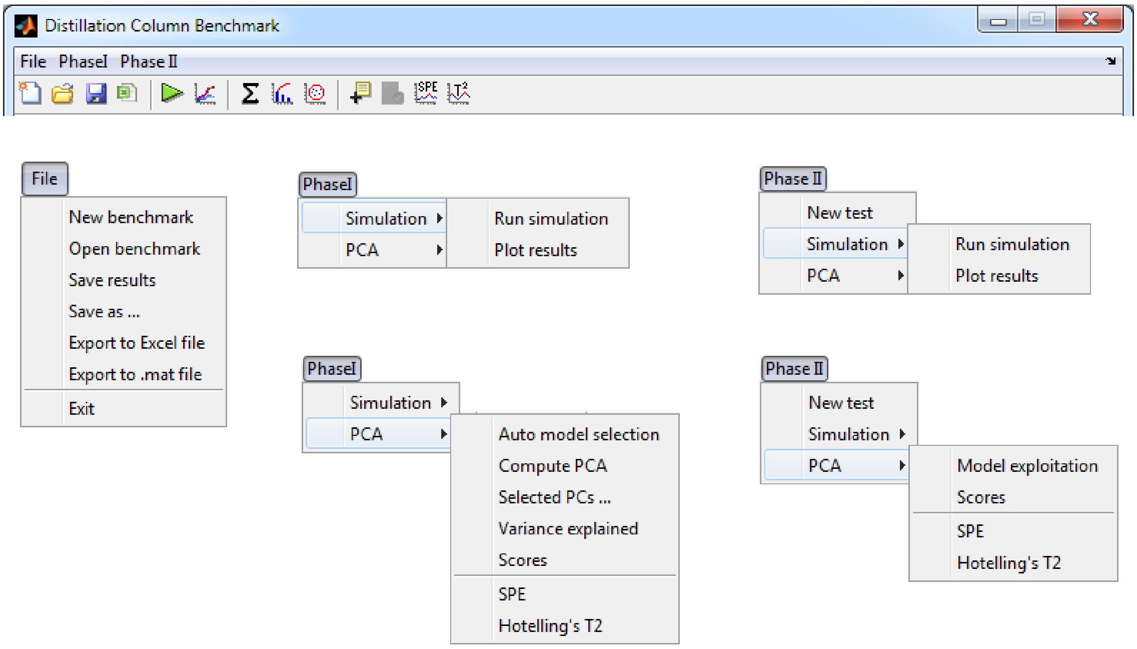


Fig. 2.- Toolbar (top) and application menus (bottom).

3.1. Phase I. Model building

3.1.1. Introduction

The main goal in Phase I is to model the in-control process performance based on a set of historical reference data collected at normal operating conditions (NOC). This data set is one in which the process has been operating consistently (stable over time) in an acceptable manner, and in which only good quality products have been obtained. Occasionally, this historical in-control data set is not directly available, but has to be extracted from historical databases in an iterative fashion. This explorative analysis of historical databases is a useful technique for improving process understanding and detecting past faults in the process (out-of-control samples). By correctly diagnosing their root causes, some countermeasures can be implemented, optimizing the future performance of the process.

Consider that the historical database consists of a set of *N* multivariate observations (objects or samples) on *K* variables (on-line process measurements, dimensional variables or product quality data) arranged in a (*NxK*) data matrix **Z**. Variables in matrix **Z** are often pre-processed by mean-centering and scaling to unit variance. With mean-centering the average value of each variable is calculated and then subtracted from the data. This usually improves the interpretability of the model because all pre-processed variables will have mean value zero. By scaling to unit variance each original variable is divided by its standard deviation and will have unit variance. Given that projection methods are sensitive to scaling; this is particularly useful when the variables are measured in different units. Other different types of scaling methods are available in the literature [(9)] (e.g., block scaling, Pareto scaling, …). After pre-processing, matrix **Z** is transformed into matrix **X**.

Principal Component Analysis (PCA) is used to reduce the dimensionality of the process by compressing the high-dimensional original data matrix **X** into a low-dimensional subspace of dimension *A* (*A* ≤ rank(**X**)), in which most of the data variability is explained by a fewer number of latent variables, which are orthogonal and linear combinations of the original ones. This is done by decomposing **X** into a set of *A* rank 1 matrices

$$\mathbf{X}=\sum_{a=1}^{A}\mathbf{t}_a\mathbf{p}_a^T+\sum_{a=A+1}^{\mathrm{rank}(\mathbf{X})}\mathbf{t}_a\mathbf{p}_a^T=\mathbf{TP}^T+\mathbf{E}=\mathbf{X}^*+\mathbf{E} \quad (6)$$

P (*KxA*) is the loading matrix containing the loading vectors $\mathbf{p}_a$, which are the eigenvectors, corresponding to the *A* largest eigenvalues of the covariance matrix of the original pre-treated data set **X**, and define the directions of highest variability of the new latent *A*-dimensional subspace.

T (*NxA*) is the score matrix containing the location of the orthogonal projection of the original observations onto the latent subspace. The columns $\mathbf{t}_a$ of the score matrix **T** ($\mathbf{t}_a = \mathbf{X}\ \mathbf{p}_a$) represent the new latent variables with variances given by their respective eigenvalues (λ_a). These new latent variables summarize the most important information of the original *K* variables, and thus can predict (reconstruct) **X** with minimum mean square error, $\mathbf{X}^* = \mathbf{TP}^T$. Matrix **E** (*NxK*) contains the residuals (statistical noise), *i*.e. the information that is not explained by the PCA model.

Eq. (6) shows that the PCA model transforms each *K*-dimensional original observation vector $\mathbf{x}_i$ (*i*-th row of matrix **X**) into an *A*-dimensional score vector $\mathbf{t}_i^T = \{t_{i1}, t_{i2}, \ldots t_{iA}\}$ (*i*-th row of matrix **T**) and a residual vector $\mathbf{e}_i$ (*i*-th row of matrix **E**).

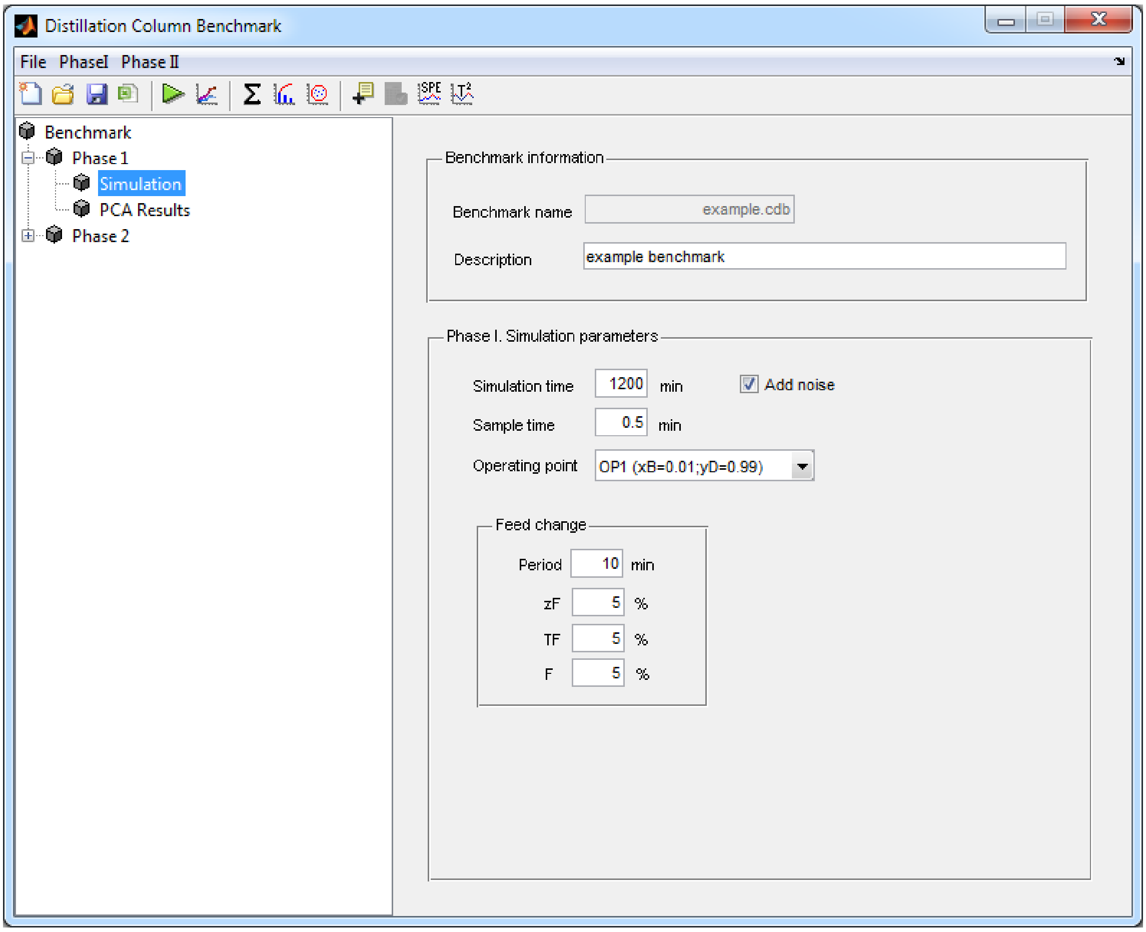
The dimension of the latent variable subspace is often quite small compared with the dimension of the original variable space (*i*.e., *A* << rank (**X**)). Several algorithms can be used to extract the principal components. For large ill-conditioned data sets it is recommended to compute the principal components sequentially via the *NIPALS* (non-iterative partial least squares) algorithm [9] and to stop based on different criteria [3,10,11]. Another advantage of *NIPALS* algorithm is that it easily handles missing data (*i*.e., observation vectors from which some variable measurements are missing). The quality of the fitted PCA model can be evaluated by computing several parameters, such as R^2, that measures the *goodness of fit*, or Q^2 that indicates the predictive capability of the model [11].

3.1.2. Phase I parameters

During Phase I user can change several parameters that will be used for the simulation of the distillation column under normal operating conditions. These data will be used to render the PCA model for later analysis.

The parameters that are available for the simulation are the following (Fig. 3):

1. *Simulation time*
2. *Add noise*: if checked, this will add random white noise to the data.
3. *Sample time*: indicates the elapsed time between measurements.
4. *Operating point*: this combo box allows selecting the operating point for the simulation. There are three possibilities that represent three different product qualities.
5. *Feed changes*:
    a. *"Period"* text box: indicates the time between changes.
    b. Change ratios (z_F, T_F, F): percentage of change in the input signals of the model (feed composition, feed temperature and feed flow).



Benchmark information

Benchmark name: example.cdb

Description: example benchmark

Phase I. Simulation parameters

Simulation time: 1200 min

☑ Add noise

Sample time: 0.5 min

Operating point: OP1 (xB=0.01;yD=0.99)

Feed change

Period: 10 min

zF: 5 %

TF: 5 %

F: 5 %

Fig. 3.- Main window. Simulation parameters (Phase I).

When the simulation is set up, the user can run it (menu option *Phase I* >> *Simulation* >> *Run simulation*). Then the Simulink model parameters are updated and the simulation starts. When the simulation finishes, results are stored and plotted in a graph (Fig.S-2). The user can access to this plot again using the “Plot results” menu option or the corresponding toolbar button.

The results plotted are the following:

z_F, feed composition

T_F, feed temperature (ºC)

F, feed volumetric flow (L/h)

L, reflux flow (L/h)

D, distillate flow (L/h)

M_D, condenser holdup (kmol)

y_D, distillate composition

V, boilup flow (L/h)

B, bottom flow (L/h)

M_B, reboiler holdup (kmol)

x_B, bottom composition

To ease the visualization of the figure, the 41 stage temperatures (T_1-T_{41}) are not plotted, but they are exported to the Excel file.

3.1.3. Model size

As stated before, the main purpose of Phase I is to fit a model from observations collected at normal operating conditions (NOC). The number of observations is proportional to the simulation time for a given sample time. The higher the number of observations under NOC, the better the model, but this increases matrices dimensions and, therefore, computing requirements. So, there must be an appropriate number of observations that renders a model able to detect abnormal situations with a minimum computational effort. In order to find this appropriate number, the percentage of points that are outside control limits is computed for several tests with different simulation times. The SPE statistic is used (see section 3.1.5) to detect if a point is in control with respect to the correlation of the model.

The methodology to determine the appropriate number of observations is based on selecting a random subset of observations to fit the PCA model and testing it against the whole dataset. The size of the random model increases gradually till it reaches an acceptable percentage of out-of-control points. The iterative process follows these steps (Fig. 4):

1. Select a random subset of observations from the dataset. The number of observations n_P in this subset is a percentage p of the total number of observations n_T in the full dataset. This percentage starts with 0.1% and increases by 0.1% if n_P is lower than 10 or n_P has not changed with respect to the previous value.

2. Fit the PCA model for the selected subset of observations.

3. Exploit the obtained model with the full set and compute the percentage of out-of-control points p_{OC} in the SPE control chart.

4. Repeat steps 1 through 3 r times to get the average p_{OC}.

5. If the average p_{OC} is lower than the threshold percentage $p_{OC,max}$ = 100 α (where α is the false alarm rate), the process stops. To do this, a one-tailed t-test for the mean at significance level 0.1 is performed. The model used for PCA monitoring will be the one with minimum number of out-of-control points. Otherwise, proceed to step 1 by increasing p by 0.1%.

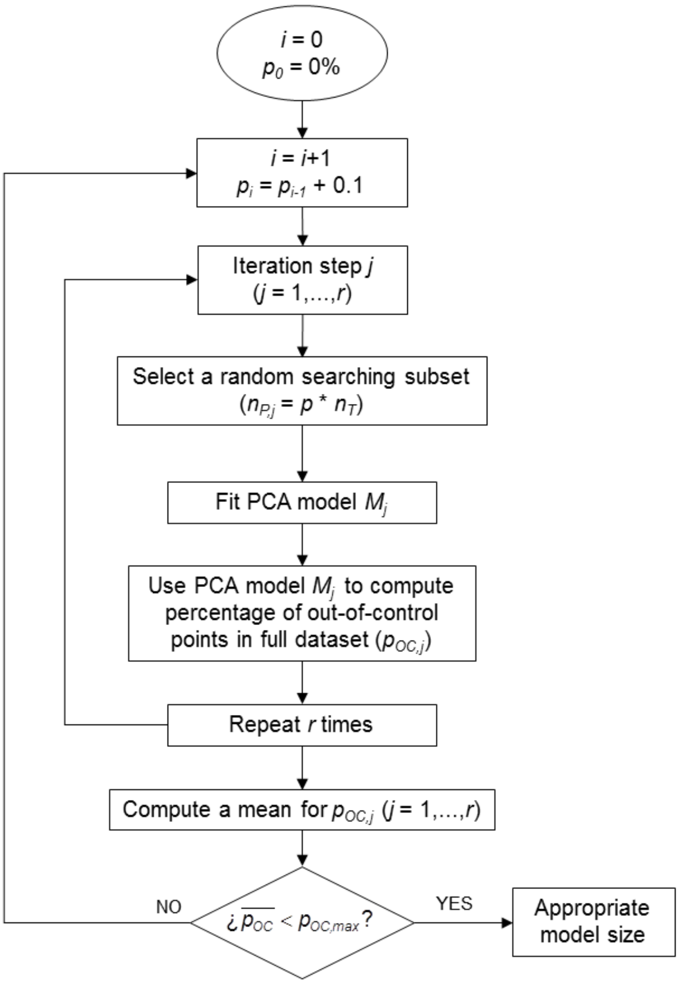

i = 0
p_0 = 0%
i = i+1
p_i = p_{i-1} + 0.1
Iteration step j
(j = 1,…,r)
Select a random searching subset
($n_{P,j}$ = p * n_T)
Fit PCA model M_j
Use PCA model M_j to compute percentage of out-of-control points in full dataset ($p_{OC,j}$)
Repeat r times
Compute a mean for $p_{OC,j}$ (j = 1,…,r)
¿$\overline{p_{OC}}$ < $p_{OC,max}$?
NO
YES
Appropriate model size

Fig. 4.- Methodology for determining the appropriate number of observations for the NOC model.

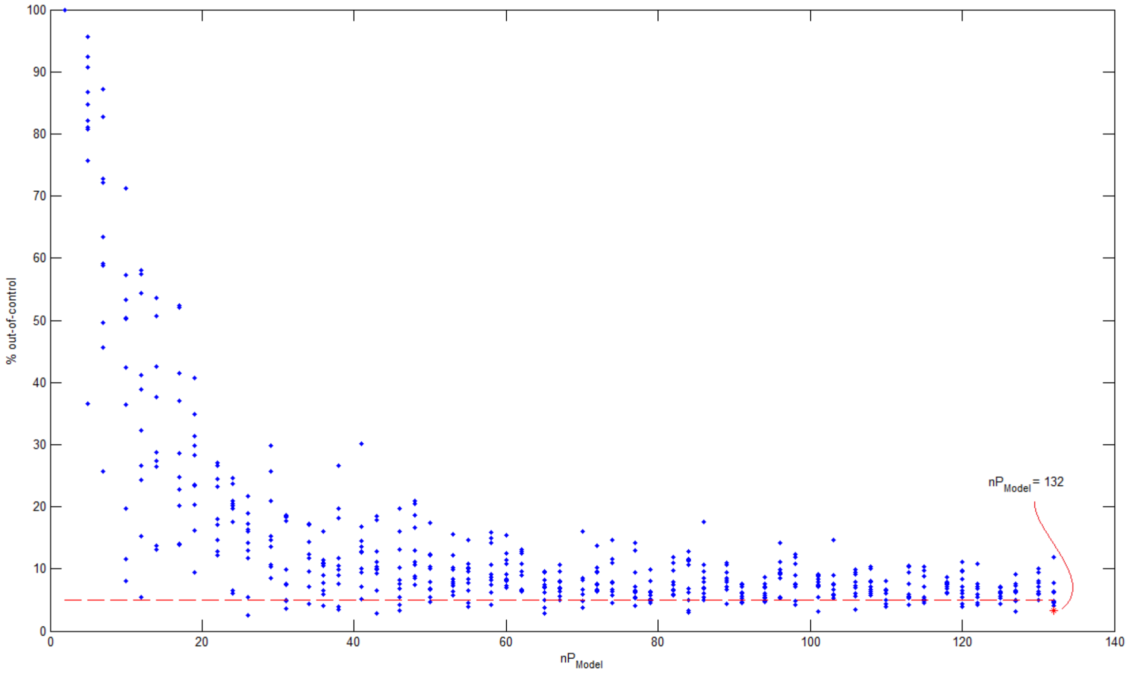


Fig. 5.- Percentage of out-of-control points in SPE chart vs. model size

The above methodology is implemented in the *Phase I* >> PCA >> *Auto model selection* menu (see Fig. 2). Fig. 5 shows the results based on a set of 2400 observations, obtained from a 1200 min simulation with a sample time of 0.5 min. The percentage of out-of-control points in the SPE chart decreases rapidly with model size, converging towards 5%. Acceptable results are obtained when model size is greater than approximately 132 observations, which means a simulation time of 66 min (for a 0.5 min sample time).

3.1.4. PCA computation

As an example, a model based on a 120 min simulation with a sample time of 0.5 min is used. Then, the data matrix for model fitting (**Z** matrix) has 240 observations (N=240). The selected variables are F, z_F, T_F, L, V, D, B, M_D, M_B, x_B, y_D and the 41 temperatures (T_1,…,T_{41}), so K=52.

PCA is computed via Singular Value Decomposition (*SVD*). Table 1 shows the source code.

Table 1.- MATLAB code for PCA using *SVD*

```
%% Compute PCA through SVD ...
[N K] = size (Z); % N observations, K variables
Zmean = mean(Z);
Zstd = std(Z);
% Standardized X matrix ...
X = (Z - repmat(Zmean,[N 1])) ./ repmat(Zstd,[N 1]);
% SVD ...
[U,S,V] = svd(X);
%Scores
T = U * S;
%Loadings
P = V;
%Eigenvalues of the covariance matrix of X
L = (diag(S) .* diag(S))/(N-1);
```

The results of the script are the following:

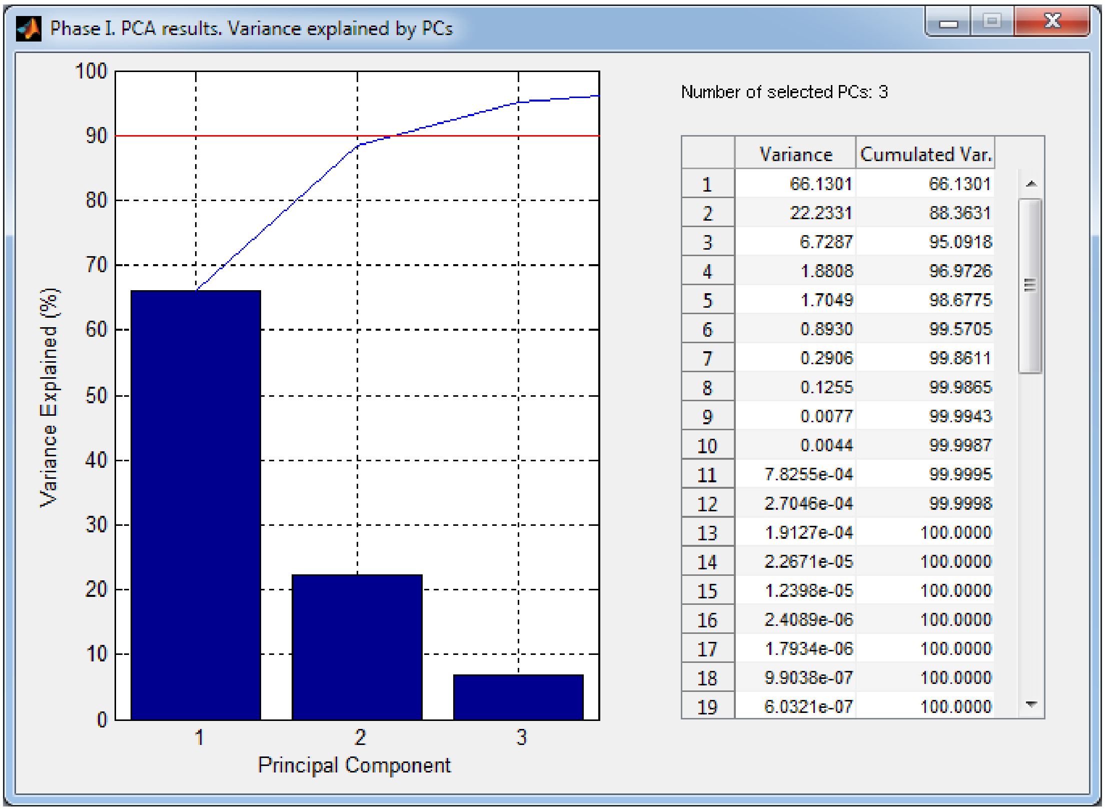
- **P**, contains the coefficients of the linear combinations of the original variables that generate the principal components (loading matrix).
- **T**, contains the coordinates of the original data in the new coordinate system defined by the principal components (score matrix).
- **L**, is a vector containing the eigenvalues of the covariance matrix of **X**.

The maximum number of components *A* extracted in a principal component analysis is equal to the number of observed variables *K* being analyzed (if *K*≤*N*). However, in most analyses, only the first few components account for meaningful amounts of variance, so only these first few components are retained, interpreted, and used in subsequent analyses. There are plenty of rules to determine the number of *PCs* to extract [3,10,11,12]. In this case, the first *A PCs* that accumulate at least 90% of explained variance are retained (see Table 2).

Table 2.- MATLAB code for *PC* selection

```
% pvar – Percentage of variance explained by each principal component
pVar = 100*L./sum(L);
% A - number of PCs that explain >90% of variance
A = find(cumsum(pVar)>=90,1,'first');
pareto(pVar);
xlabel('Principal Component');
ylabel('Variance Explained (%)');
```

In this case, a 95.09 % of variance explained is reached by the third *PC*, so *A*=3. Fig. 6 shows a Pareto chart with the percentage of variance explained by each *PC*. This plot is obtained through the *Phase I >> PCA >> Variance explained* option menu.



Number of selected PCs: 3

	Variance	Cumulated Var.
1	66.1301	66.1301
2	22.2331	88.3631
3	6.7287	95.0918
4	1.8808	96.9726
5	1.7049	98.6775
6	0.8930	99.5705
7	0.2906	99.8611
8	0.1255	99.9865
9	0.0077	99.9943
10	0.0044	99.9987
11	7.8255e-04	99.9995
12	2.7046e-04	99.9998
13	1.9127e-04	100.0000
14	2.2671e-05	100.0000
15	1.2398e-05	100.0000
16	2.4089e-06	100.0000
17	1.7934e-06	100.0000
18	9.9038e-07	100.0000
19	6.0321e-07	100.0000

Fig. 6.- Variance explained by the principal components

User can change the number of selected *PCs* through the *Phase I >> PCA >> Selected PCs …* option menu that will show an input dialog to enter the number of *PCs* desired by the user.

3.1.5. PCA statistics

According to Eq. (6), the residual matrix **E** is computed in the following way (note that only the first *A* elements in both loading and score matrices are used):

```
Xstar = T(:,1:A)*P(:,1:A)';
E = X - Xstar;
```

The residuals of the original data (**Z** matrix) are computed through the reconstructed original data (**Z*** matrix), which is obtained from the above reconstructed standardized data (**X*** matrix):

```
Zstar = Xstar .* repmat(Zstd,[N 1]) + repmat(Zmean,[N 1]);
EZ = Z - Zstar;
```

From the scores and the residuals (prediction errors) associated with each observation, two complementary (orthogonal or independent) statistics are derived: the SPE (sum of squared prediction errors) and the Hotelling's T_A^2 .

The SPE statistic for *i*-th observation $\mathbf{x}_i$ is given by

$$SPE = \mathbf{e}_i^T \mathbf{e}_i = \left(\mathbf{x}_i - \mathbf{x}_i^*\right)^T \left(\mathbf{x}_i - \mathbf{x}_i^*\right) \quad (7)$$

where $\mathbf{e}_i$ is the residual vector of *i*-th observation, and $\mathbf{x}_i^*$ is the prediction of the observation vector $\mathbf{x}_i$ from the PCA model. The SPE statistic represents the squared Euclidean (perpendicular) distance of an observation from this subspace, and gives a measure of how close the observation is from the *A*-dimensional subspace.

Table 3.- MATLAB code for SPE statistic computation

```
% Sum of squared prediction errors
SPE = diag(E()*E()');
% UCL for SPE at false alarm rate alpha (Phase I)
alpha = 0.05;
SPEmean = mean(SPE);
SPEvar = var(SPE);
ChiSquared = chi2inv(1-alpha,2*SPEmean^2/SPEvar);
SPE_UCL = SPEvar*ChiSquared/(2*SPEmean);
```

On the other hand, the T_A^2 statistic for the *i*-th observation is defined as [(2)]

$$T_{\mathrm{A}}^2 = \mathbf{t}_i^T \boldsymbol{\Theta}^{-1} \mathbf{t}_i = \sum_{\mathrm{a}=1}^{\mathrm{A}} \frac{t_{\mathrm{a}}^2}{\lambda_{\mathrm{a}}} \quad (8)$$

where $\boldsymbol{\Theta}(\mathrm{A} \times \mathrm{A})$ is the covariance matrix of **T** (diagonal matrix of the highest *A* eigenvalues $\{\lambda_1,\ldots,\lambda_A\}$). This is the Hotelling-T^2 statistic when a reduced subspace with *A* components is used instead of the original variables space, and it represents the estimated squared Mahalanobis distance from the center of the latent subspace to the projection of an observation onto this subspace. Table 4 shows the MATLAB code that computes *Tsquared*, a vector of dimension *N* that contains T_A^2 for each observation in matrix **X**.

Table 4.- MATLAB code for *Hotelling-T^2* statistic computation

```
% Hotelling T2 computation for each observation:
Tsquared = zeros(1,N);
for i=1:N;
    for a = 1:A;
        Tsquared(i) = Tsquared(i) + T(i,a)^2 / L(a);
    end;
end;
% UCL for Hotelling T2 at false alarm rate alpha (Phase I)
alpha = 0.05;
F = finv(1-alpha,A,N-A-1);
B = (A/(N-A-1))*F/(1+(A/(N-A-1))*F);
UCL_T2 = (N-1)^2*B/N;
```

3.1.6. Multivariate control charts

From the above two statistics, in PCA-based MSPC two complementary multivariate control charts are built. Shewhart-type control charts for individual observations are often used in practice. The control limits of the multivariate control charts are calculated following the traditional SPC philosophy. As commented in 3.1.1, in Phase I, an appropriate historical or reference set of data (collected from one or various periods of plant operation when performance was good) is chosen which defines the normal or in-control operating conditions (NOC) for a particular process corresponding to common-cause variation. The in-control PCA model is then built on these data. Any periods containing variations arising from special events that one would like to detect in the future are omitted at this stage. The choice of the reference (in-control) data set is critical to the successful application of the procedure [(14)]. Control limits for good operation on the control charts are defined based on this reference data set. In Phase II, values of future measurements are compared against these limits.

Several procedures can be used for calculating upper control limits (UCL) for the Shewhart SPE chart at the false alarm rate (type I risk) α. In this GUI, an approximation based on the weighted chi-squared distribution ($g\chi_h^2$) proposed by Box [(15)] is used. Nomikos and MacGregor [(16)] suggested a simple and fast way to estimate the parameters *g* and *h* that is based on matching moments between a $g\chi_h^2$ distribution and the sample distribution of SPE. The mean ($\mu = gh$) and variance ($\sigma^2 = 2g^2h$) of the $g\chi_h^2$ distribution are equated with the sample mean (*h*) and variance (*v*) of the SPE sample. Hence, the upper SPE control limit at false alarm rate α is given by

$$\mathrm{UCL}(\mathrm{SPE})_\alpha = \frac{\mathrm{v}}{2\mathrm{b}}\chi^2_{\left(2\mathrm{b}^2/\mathrm{v}\right),\alpha} \quad (9)$$

where $\chi^2_{\left(2\mathrm{b}^2/\mathrm{v}\right),\alpha}$ is the 100 (1-α) % percentile of the corresponding chi-squared distribution.

Assuming that the scores follow a multivariate normal distribution, upper control limits (UCL) for the Shewhart T_A^2 chart at false alarm rate (type I risk) α can be obtained for Phase I by the following equation:

$$\mathrm{UCL}\left(T_A^2\right) = \frac{(\mathrm{N}-1)^2}{\mathrm{N}} B_{(\mathrm{A}/2,(\mathrm{N}-\mathrm{A}-1)/2),\alpha} \quad (10)$$

where $B_{(\mathrm{A}/2,(\mathrm{N}-\mathrm{A}-1)/2),\alpha}$ is the 100(1-α)% percentile of the corresponding beta distribution that can be computed from $F_{(A,N-A-1),\alpha}$, i.e. the 100(1-α)% percentile of the corresponding *F* distribution, by using the following relationship [(13)]

$$B_{(\mathrm{A}/2,(\mathrm{N}-\mathrm{A}-1)/2),\alpha} = \frac{\left(\mathrm{A}/(\mathrm{N}-\mathrm{A}-1)\right)F_{(\mathrm{A},\mathrm{N}-\mathrm{A}-1),\alpha}}{\left(1+\left(\mathrm{A}/(\mathrm{N}-\mathrm{A}-1)\right)F_{(\mathrm{A},\mathrm{N}-\mathrm{A}-1),\alpha}\right)} \quad (11)$$

For Phase II, the corresponding UCL is given by

$$\mathrm{UCL}\left(T_A^2\right)_\alpha = \frac{\mathrm{A}\left(\mathrm{N}^2-1\right)}{\mathrm{N}\left(\mathrm{N}-\mathrm{A}\right)} F_{(\mathrm{A},(\mathrm{N}-\mathrm{A})),\alpha} \quad (12)$$

The difference in both control limits comes from the fact that in Phase I, the same observation vectors **x**ᵢ collected in the reference data set are used for two purposes: (*i*) to build the PCA model and work out the control limits of the charts, and (ii) to check whether they fall within these control limits. Therefore, observations in the reference data set are not independent of PCA model parameters used to derive the statistics to be monitored. In contrast, in Phase II new observations (not used for model building) are checked against the control limits calculated from the in-control data, and therefore, independence is guaranteed. Anyway, if a large reference data set is available Eq. (12) an also be used for estimating the control limits in Phase I.

On the other hand, the GUI can also display two dimensional (2-D) score plots, which represent the projection of the observations onto the plane defined by two given scores. The confidence interval of the score plot is computed based on the Hotelling T^2 statistic. The normalized (1- α) confidence region for a two dimensional score plot of dimension *a* and *b* is given by the following ellipse:

$$\left(\frac{\mathrm{t}_1}{\mathrm{axis}_\mathrm{a}}\right)^2 + \left(\frac{\mathrm{t}_2}{\mathrm{axis}_\mathrm{b}}\right)^2 = 1 \quad (13)$$

with axis:

$$\boldsymbol{axis}_{\boldsymbol{c}} = \pm\sqrt{s_{t_c}^2 \times (F_{2,N-2})_\alpha \times 2 \times \frac{N^2-1}{N(N-2)}}; \quad \mathbf{c=a\ or\ b} \quad (14)$$

where $s_{\mathrm{t_c}}^2$ is the variance of the score t_a (or t_b), *i*.e., the eigenvalue associated to the corresponding eigenvector of the covariance matrix of **X**:

$$s_{\mathrm{t_c}}^2 = \mathrm{var}(t_\mathrm{c}) = \lambda_\mathrm{c} \quad (15)$$

where λ_c represents the *c*-th element of vector **L**.

Table 5.- MATLAB code for the 2-D score plot.

```
% Score scattered plot for the first two components
plot(T(:,1),T(:,2),'+')
xlabel('1st Principal Component')
ylabel('2nd Principal Component')
% Confidence region:
alpha = 0.05;
F = finv(1-alpha,2,N-2);
axisa = sqrt(L(1)*F*2*(N^2-1)/(N*(N-2)));
axisb = sqrt(L(2)*F*2*(N^2-1)/(N*(N-2)));
rectangle('Position',[-axisa,-axisb,2*axisa,2*axisb],'Curvature',[1,1], ...
          'EdgeColor','r');
```

The user can plot the SPE chart and the T^2 chart in Phase I (Fig. 7) using the corresponding menu options in the *Phase I >> PCA* submenu or by clicking the corresponding toolbar buttons. Both control charts have two upper control limits: one for a confidence level 1-α=0.95 (red continuous line) and another for 1-α=0.99 (red dash line).

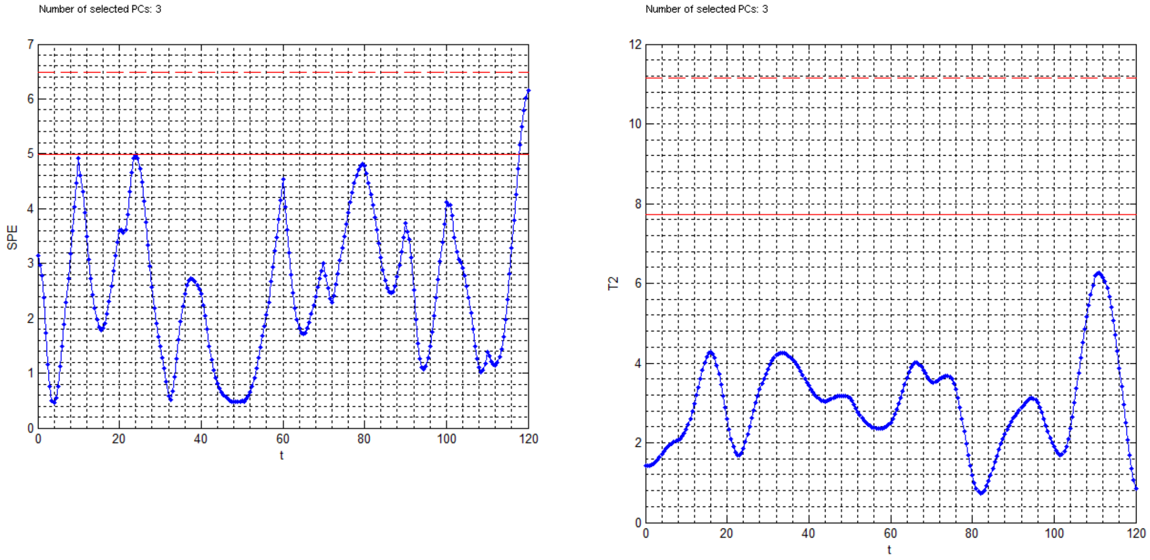


Fig. 7.- SPE and T^2 charts (Phase 1)

There are some points slightly out of control limits of the SPE and T_A^2 Shewhart chart but this is an acceptable situation, because a determined number of points are expected to slightly exceed the upper control limit given the in-control model. For a confidence level 1- α =0.95, this expected number of out-of-control points is computed as *n_{obs} x 0.05 = 240 x 0.05 = 12*

Fig. 8 shows a plot of the original observations projected onto the latent subspace formed by the first two principal components (see MATLAB code in Table 5). This plot is obtained through the *Phase I >> PCA >> Scores* menu. The plot shows two confidence regions: one region at confidence level $1-\alpha$ = 0.95 (red continuous line) and another for $1-\alpha$ = 0.99 (red dash line).

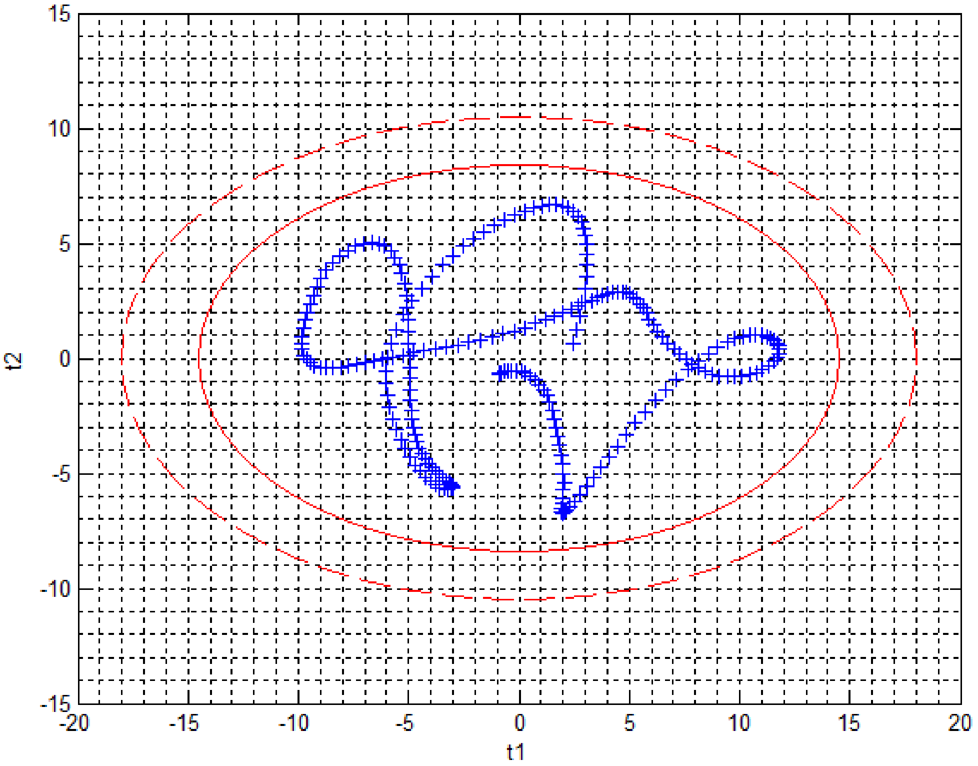


Fig. 8.- Scatter plot for the two first components.

3.2. Phase II. Model exploitation

3.2.1. Introduction

Once the reference PCA model and the control limits for the multivariate control charts are obtained, new process observations can be monitored on-line. When a new observation vector $\mathbf{z}_i$ is available, it is pre-processed and projected onto the PCA model yielding the scores and the residuals, from which the SPE and the Hotelling's T_A^2 values are calculated. This way, the information contained in the original *K* variables is summarized in these two indices that are plotted in the corresponding multivariate SPE and T_A^2 control charts. No matter what the number of the original variables *K* is, only two points have to be plotted on the charts and checked against the control limits. The SPE chart should be checked first. If the points remain below the control limits in both charts the process is considered to be in control. If a point is detected to be beyond the limits of one of the charts, then a diagnostic approach to isolate the original variables responsible for the out-of-control signal is needed. In PCA-based MSPC, one of the most widely used approaches is the contribution plots [(14)]. Contribution plots are a powerful tool for fault diagnosis. They provide a list of the process variables that contribute numerically to the out-of-control condition, but they do not reveal the actual cause of the fault. Those variables and any variable highly correlated with them should be investigated. Incorporation of technical process knowledge is crucial to diagnose the problem and discover the root causes of the fault.

3.2.2. Failure scenario simulation

There is only one model for each benchmark but the user can generate several scenarios to simulate different disturbances and failures. This could be done by creating new test sets in Phase II, through the corresponding *Phase II >> New test* menu option or button toolbar. Then the application adds a new node to the “Phase II” tree.

When the user selects a “Test #…” child node for “Phase II” in the tree view, the window corresponding to the parameters that will be used for the troubleshooting simulations is shown in Fig. 10:

1. *Simulation time*
2. *Activate disturbance* in z_F, T_F or F.

   When checked, it will enable several controls to specify the disturbance:

   a. *At time*: time at which the disturbance will be activated.

   b. *Type*: three types are available: spike, ramp and pulse.

   c. *Size*: indicates the percentage of variation used for the disturbance signal.

   d. *Duration*: for spike and ramp signals, the user only has to specify the first parameter that is the duration of the whole signal. For pulse signal, a second parameter is needed to indicate the duration of the steady part of the signal (see Fig. 9).

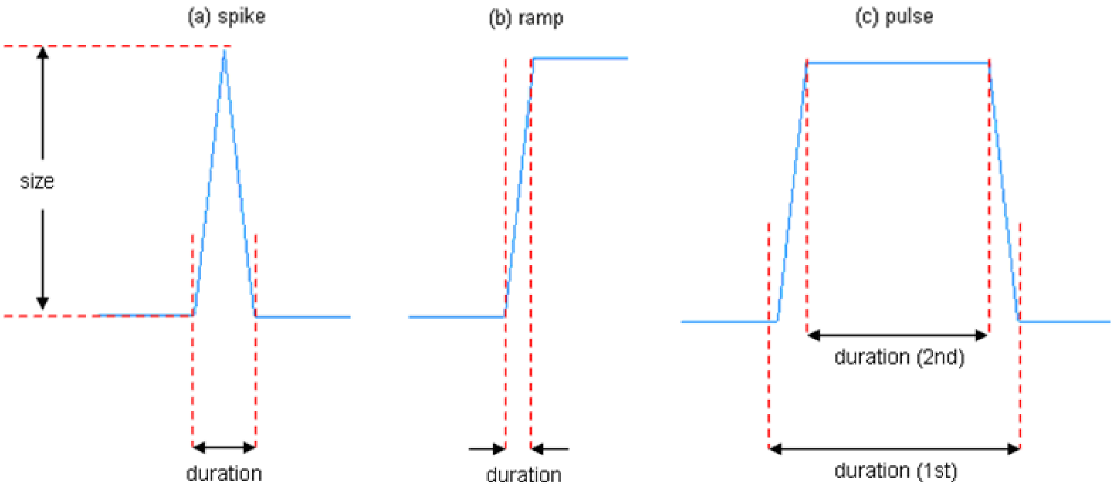
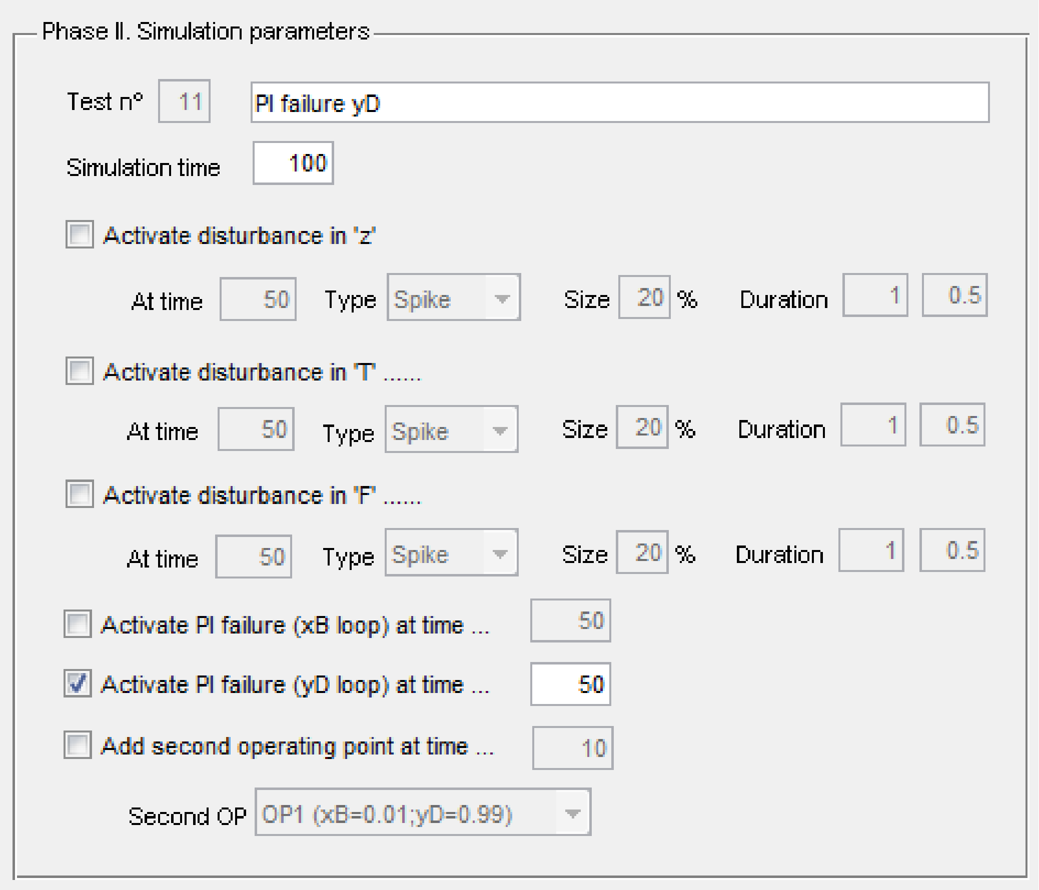


Fig. 9.- Types of disturbances signals.

3. *Activate PI failure* (x_B loop or y_D loop). This simulates a failure in the PI control loop. The user has to specify the time at which the failure will take place. The Simulink model then modifies the flow value returned by the control loop to simulate the failure.

4. *Add second operating point*. This is used for transitions between operating points. If checked, the combo box labeled “Second OP” and the text box “… at time” will be enabled to specify the second operating point and when the transition will take place.

The rest of the parameters are the same that those used for the simulation in Phase I.
When the simulation parameters are set up the user can start the simulation in the same way as in Phase I. The results will be saved and a plot will be shown.



Phase II. Simulation parameters

Test nº 11 PI failure yD

Simulation time 100

□ Activate disturbance in 'z'

At time 50 Type Spike Size 20 % Duration 1 0.5

□ Activate disturbance in 'T' ......

At time 50 Type Spike Size 20 % Duration 1 0.5

□ Activate disturbance in 'F' ......

At time 50 Type Spike Size 20 % Duration 1 0.5

□ Activate PI failure (xB loop) at time ... 50

☑ Activate PI failure (yD loop) at time ... 50

□ Add second operating point at time ... 10

Second OP OP1 (xB=0.01;yD=0.99)

Fig. 10.- Phase II simulation parameters.

3.2.3. SPE contribution plot

When an out-of-control situation is detected on the SPE plot, the contribution of each variable of the original data set is simply given by

$$\mathrm{Cont}\left(\mathrm{SPE};\mathbf{x}_{\mathrm{new},k}\right)=\mathrm{sign}\left(\mathbf{e}_{\mathrm{new},k}\right)\mathbf{e}_{\mathrm{new},k}^{2}=\mathrm{sign}\left(\mathbf{e}_{\mathrm{new},k}\right)\left(\mathbf{x}_{\mathrm{new},k}-\mathbf{x}_{\mathrm{new},k}^{*}\right)^{2} \quad (16)$$

When the user selects an observation in the SPE chart (see left plot in Fig. 11), the GUI shows a bar plot with the contributions (right plot in Fig. 11). Then, variables with high contributions (in absolute value) should be investigated.

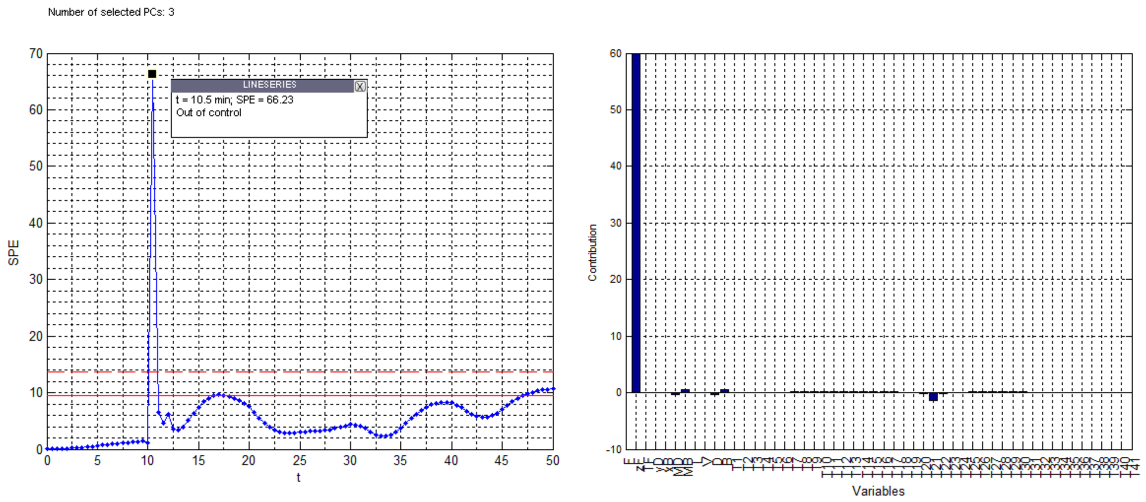


Fig. 11.- SPE chart (left) and SPE contribution plot (right).

Table 6.- MATLAB code for SPE contribution plot

```
% Standardized X (phase II)
Xnew = (Znew - repmat(Zmean,[Nnew 1])) ./ repmat(Zstd,[Nnew 1]);
% Scores for the new observations
Tnew = Xnew * P;
% Model prediction for new observations
Xstarnew = Tnew(:,1:A) * P(:,1:A)';
% Prediction error for new observations
Enew = Xnew - Xstarnew;
% SPE contribution plot for idObs observation
bar((EnewSquared(idObs,:).*sign(Enew(idObs,:))));
```

3.2.4. Scores contribution plot

If the abnormal observation is detected by the T_A^2 chart the diagnosis procedure is carried out in two steps:

1. *A* bar plot of the normalized scores for that observation $\left(t_{new,a}^2 / \lambda_a\right)$ is plotted and the *a*-th score with the highest normalized value is selected.
2. The contribution of each original *k*-th variable to this *a*-th score at this new abnormal observation is given by

$$\text{Cont}\left(t_{new,a} ; x_{new,k}\right) = p_{ak}\, x_{new,k} \quad (17)$$

where p_{ak} is the loading of the *k*-th variable at the *a*-th component.

When the user selects an observation in the T^2 chart (see left plot in Fig. 12), the GUI shows a bar plot with the normalized scores (upper right plot in Fig. 12) and automatically selects the highest normalized score, showing the corresponding scores contribution plot (lower right plot in Fig. 12). Positive values (for both scores and contributions) are colored in green and negative values, in red. Variables on the contribution plot with high values but with the same sign (color) as the score should be investigated (contributions of the opposite sign, will only make the score smaller).

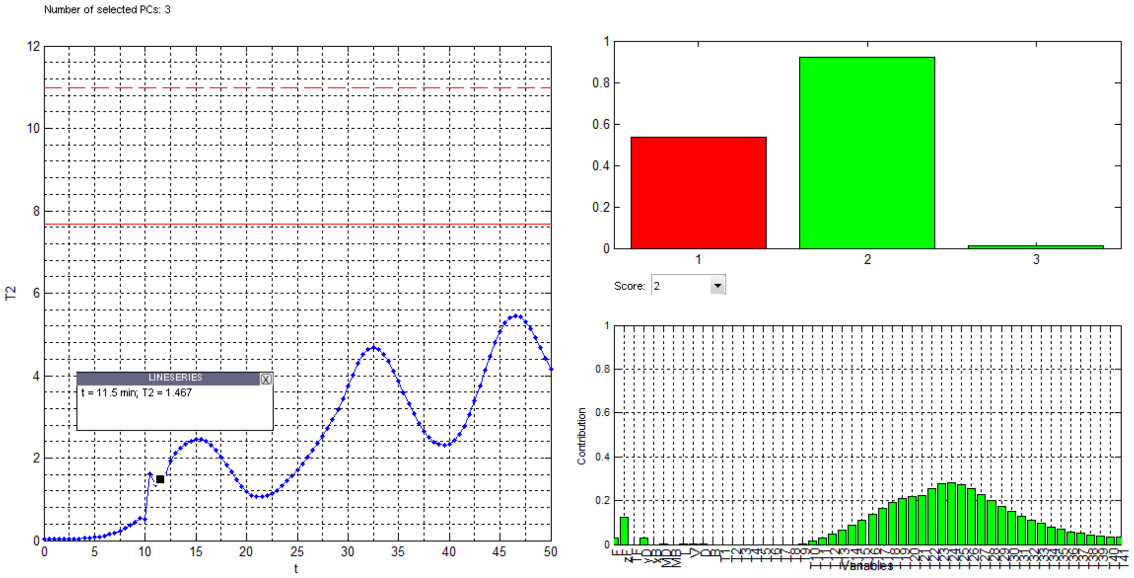


Fig. 12.- T^2 chart (left), normalized scores (upper right) and scores contribution plot (lower right).

Table 7.- MATLAB code for scores contribution plot

```
% Standardized X (phase II)
Xnew = (Znew - repmat(Zmean,[Nnew 1])) ./ repmat(Zstd,[Nnew 1]);
% Scores for the new observations
Tnew = Xnew * P;
% Model prediction for new observations
Xstarnew = Tnew(:,1:A) * P(:,1:A)';
% Prediction error for new observations
Enew = Xnew - Xstarnew;
TsquaredNew = zeros(1,Nnew);
for i=1:Nnew;
    for a = 1:A;
        TsquaredNew(i) = TsquaredNew(i) + Tnew(i,a)^2 / L(a);
    end;
end;
```

```
% Phase II.Contribution to Hotelling's T2
% If the abnormal observation is detected by the T2 chart the diagnosis procedure is carried out in two steps:
%     1) A bar plot of the normalized scores (t2/L) for that observation is plotted and the 'a' score with the highest normalized value is selected.
NormScores = T(idObs,1:A).^2 ./ L(1:A)';
bar(NormScores);
%    2) Then, the contribution of each original 'k' variable to this 'a' score at this new abnormal observation is given by
for k=1:size(P)
    cont(k) = P(k,a) * X(idObs,k);
end
bar(cont);
```

4. Results

4.1. Benchmark simulations

The benchmark generated for the example is the following:

Table 8.- Phase I parameters for benchmark#1

Parameter	Value
Simulation time	1200 min
Sample time	0.5 min
Add noise	Yes
Operating point	OP 1: x_B=0.01; y_D=0.99
Feed changes	Period: 10 min Change ratios: z_F = 5%, F = 5%, T_F = 5%

The *Phase II >> PCA >> Model exploitation* menu option uses the model built in Phase I to monitor the simulated tests in Phase II. Then, the above statistics SPE and Hotelling's T^2 are computed and shown in charts that are used for process monitoring.

First of all, a 1200 min simulation with no disturbances (that is, under normal operating conditions, NOC) is run. Results are shown in Fig. S-2. The *Phase I* >> PCA >> *Auto model selection* menu is used to get a simpler model (see 3.1.3)

Then, several types of signal disturbances and failures in regulatory controls are designed to check the PCA monitoring capabilities of the model obtained in Phase I (see Table 9). The column "Signal duration" for pulse signal type has the format *x (y)*, where *x* is the whole duration of the signal and *y* is the duration of the steady part.

Table 9.- Phase II tests parameters.

Test	Simulation time	Disturbance variable	Disturbance time	Signal	Signal duration
1	50	z_F	10	spike	1
2	50	z_F	10	ramp	1
3	50	z_F	10	pulse	1 (0.5)
4	50	T_F	10	spike	1
5	50	T_F	10	ramp	1
6	50	T_F	10	pulse	1 (0.5)
7	50	F	10	spike	1
8	50	F	10	ramp	1
9	50	F	10	pulse	1 (0.5)
10	100	PI failure (x_B)	50	-	-
11	100	PI failure (y_D)	50	-	-
12	200	Transition to operating point 2	100	-	-
13	200	Transition to operating point 3	100	-	-
14	50	z_F T_F	10 10	pulse pulse	1 (0.5) 1 (0.5)
15	50	z_F F	10 10	pulse pulse	1 (0.5) 1 (0.5)
16	50	T_F F	10 10	pulse pulse	1 (0.5) 1 (0.5)

For example, test#1 is a 50 min simulation with a spike-type disturbance in feed composition (z_F variable) at 10 min. As stated before, the SPE control chart, which measures the distance to the model, is checked first. If there is some point out of control, then the responsible variables are studied with the contribution plot. If there is not, then the T^2 control chart must be checked, which measures if the projected observations are in the zone defined by normal operating conditions. For example, for the above test#1 there are some out-of-control points in the SPE control chart (Fig. 13, left). The red line is the upper control limit at confidence level 1-α = 0.95 and the red dashed line corresponds to the upper control limit at confidence level 1-α = 0.99. When the user selects a point in the SPE chart then a contribution plot of the variables for that observation is shown (Fig. 13, right), which is useful to assess the cause of the failure. In this case, the variable that contributes greatly to the out-of-control observation is variable z_F (feed composition) which is perfectly coherent with the simulated disturbance, that is, the spike signal disturbance at 10 min of simulation. Once a point is selected, it is possible to use the arrow keys to move forward or backward and check the evolution of the contributions of each variable over time.

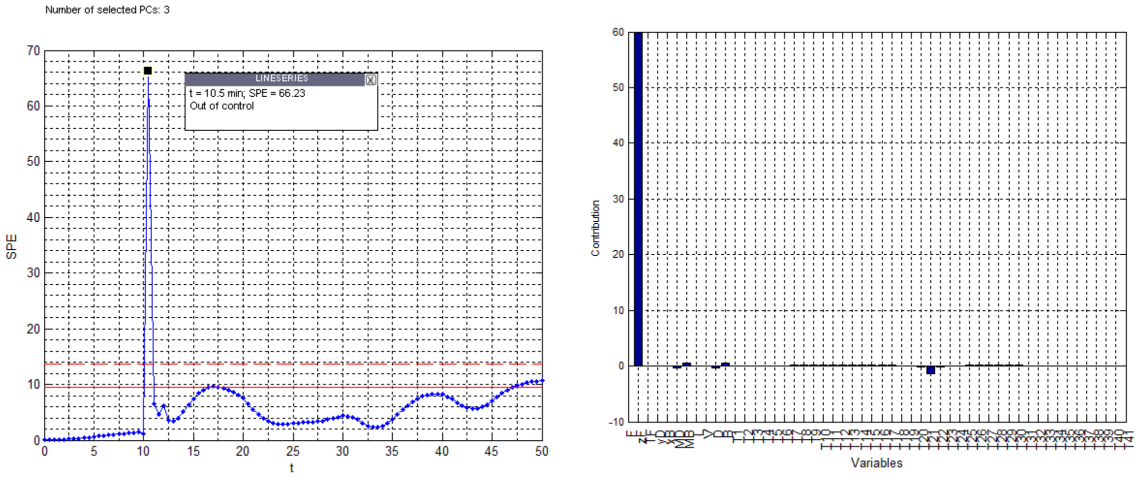


Fig. 13.- Test #1 (z-spike). SPE chart (left) and SPE contribution plot (right).

The monitoring study of the changes in feed parameters (*F*, z_F and T_F) is extended with several types of signals (see tests #1 to #9). All of these tests can be monitored with the SPE chart and its corresponding failure analysis can be done with the contribution plots (see Supplementary Material S-3).

When the signal is a ramp (see Fig. 9 for signal types), the disturbance remains at a high level and this is reflected in the SPE chart (see Fig.S-3). The contribution plot correctly diagnoses the faulty variables.

Pulse signals are also detected in SPE chart. Fig.S-4 shows results of test #9, a 50 min simulation with a pulse signal at 10 min for feed flow (*F* variable).

Initially, the cause of the failure is the high value of feed flow itself but other variables are affected greatly after this disturbance. The contribution plot in Fig.S-5 shows how the bottom holdup M_B and bottom flow *B* are outside their normal operating values due to the disturbance in feed flow, until the process reaches the steady state after 7 min after the disturbance, approximately. A high feed flow increases the bottom holdup M_B so the control loop increases the bottom flow *B* to avoid column flooding.

Test#1 has been repeated with a 15% disturbance size (Fig.S-6) and another one of 10% (Fig.S-7). Although the SPE value for the out-of-control point diminishes for those new tests, it is still detectable for a 10% disturbance size. This gives an idea of the “sensitivity” of the PCA model that was obtained with 5% random disturbances (see Table 8).

Another types of failure simulated in this paper are related to regulatory controls such the PI for composition control in x_B and y_D. They try to simulate the effect of a control valve malfunction, by limiting the flow in *V* (or *L*) to the mean value under normal conditions. For test#10 (Fig. S-8), the flow *V* is limited to a value smaller than the one needed to keep x_B under control. Due to column interactions, this affects not only the directly related variables (bottom flow *B* and x_B composition), but also distillate variables *D* and y_D. In fact, the corresponding SPE chart (Fig.S-9) shows a high contribution of *B* and x_B to the selected out-of-control point, but also a high contribution of *D* and y_D, in a lesser extent. It also shows how variable *T1* (bottom temperature) is directly related to x_B and *T41* (top temperature) is directly related to y_D.

A similar failure is simulated for the y_D control loop (Fig.S-10). In this case the limited flow is *L*, which affects flows and compositions both in bottom and distillate, as shown in the corresponding SPE chart (Fig.S-11).

Test #12 and #13 deal with transition issues. Test #12 simulates the transition from operating point 1 (x_B = 0.01, y_D = 0.99) to 2 (x_B = 0.01, y_D = 0.96) and test #13 simulates the transition from operating point 1 to 3 (x_B = 0.05, y_D = 0.99).

Fig.S-12 shows the scatter plot for the first two components in phase II for test#12. There are two clusters corresponding to the two operating regions: the first one is at (0,0) coordinates and corresponds to the first operating point and it is explained by the model developed in Phase I. The other cluster is the second operating point that is far away from our model. The contribution plot for a point of the second cluster (Fig.S-13) shows the great contribution of the temperatures of the last stages of the distillation column. These temperatures are directly related to y_D composition that has changed from 0.99 to 0.96 and, thus, the temperatures have changed accordingly.

In both tests (#12 and #13), both control charts SPE and T^2 can detect the abnormal situation (Fig.S-14). The model obtained during Phase I is not valid to explain the process variability for other operating points. Fig.S-14 also shows the high values for both statistics due to the great difference between both situations.

The last group of simulations deals with the capability of PCA monitoring to explain the causes of failures from more complicated situations that the ones presented in test#1 to #9. Here, some combinations of feed disturbances simultaneously are made (see tests#14 to #16 in Table 9). The PCA model is able to detect not only the abnormal situation but its root causes (Fig.S-15, Fig.S-16 and Fig.S-17). When *F* disturbance is present, there is a collateral effect on variable *B* (pointed out by a red circle in Fig.S-16). This situation is the same as the one shown in Fig. S-5.

4.2. External datasets

The software also incorporates the possibility to analyze external datasets. The details of the procedure to be followed are explained in the following example.

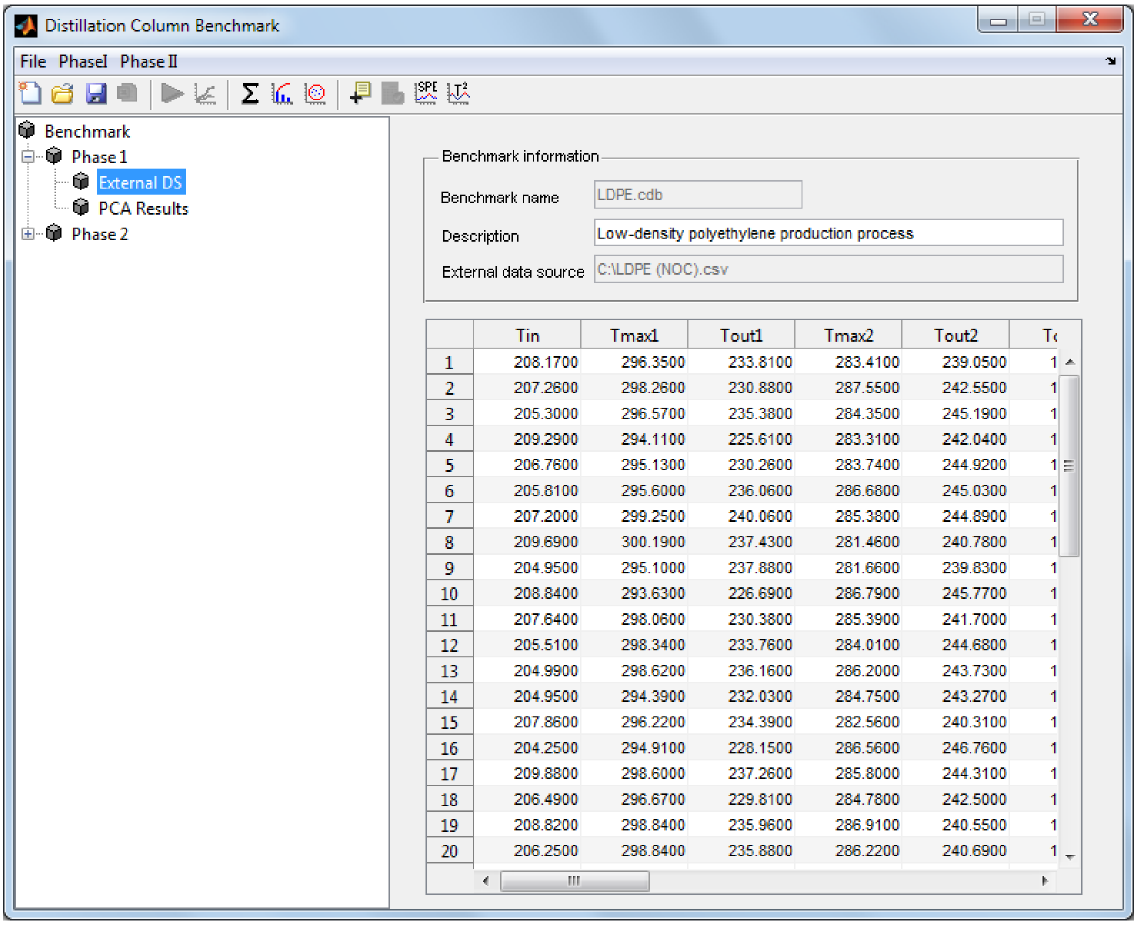
4.2.1. LDPE process

The first example corresponds to a simulation of a low-density polyethylene (LDPE) production process [18,19]. There are 14 process variables and 5 quality variables (see Table 10). The dataset used to build the PCA model (Phase I) has 50 observations from Normal Operating Conditions (NOC) and one observation with abnormal conditions. The dataset for model exploitation (Phase II) has 4 observations of a process fault developing: the impurity level in the ethylene feed in both zones is increasing.

Table 10.- Variables of the “LDPE” dataset

Variable	Description
Tin	inlet temperature to zone 1 of the reactor [K]
Tmax1	maximum temperature along zone 1 [K]
Tout1	outlet temperature from zone 1 [K]
Tmax2	maximum temperature along zone 2 [K]
Tout2	outlet temperature from zone 2 [K]
Tcin1	temperature of inlet coolant to zone [K]
Tcin2	temperature of inlet coolant to zone 2 [K]
z1	percentage along zone 1 where Tmax1 occurs [%]
z2	percentage along zone 2 where Tmax2 occurs [%]
Fi1	flow rate of initiators to zone 1 [g/s]
Fi2	flow rate of initiators to zone 2 [g/s]
Fs1	flow rate of solvent to zone 1 [% of ethylene]
Fs2	flow rate of solvent to zone 2 [% of ethylene]
Press	pressure in the reactor [atm]
Conv	quality variable: cumulative conversion
Mn	quality variable: number average molecular weight
Mw	quality variable: weight average molecular weight
LCB	quality variable: long chain branching per 1000 C atoms
SCB	quality variable: short chain branching per 1000 C atoms

To use an external dataset, choose the “File >> New benchmark” menu option and answer “Yes” to the question “Do you want to use external data?”. Then, user will be asked to select a CSV file. The data will be showed in a table under the name and project description (see Fig. 14).



Benchmark information

Benchmark name: LDPE.cdb

Description: Low-density polyethylene production process

External data source: C:\LDPE (NOC).csv

	Tin	Tmax1	Tout1	Tmax2	Tout2	Tc
1	208.1700	296.3500	233.8100	283.4100	239.0500	1
2	207.2600	298.2600	230.8800	287.5500	242.5500	1
3	205.3000	296.5700	235.3800	284.3500	245.1900	1
4	209.2900	294.1100	225.6100	283.3100	242.0400	1
5	206.7600	295.1300	230.2600	283.7400	244.9200	1
6	205.8100	295.6000	236.0600	286.6800	245.0300	1
7	207.2000	299.2500	240.0600	285.3800	244.8900	1
8	209.6900	300.1900	237.4300	281.4600	240.7800	1
9	204.9500	295.1000	237.8800	281.6600	239.8300	1
10	208.8400	293.6300	226.6900	286.7900	245.7700	1
11	207.6400	298.0600	230.3800	285.3900	241.7000	1
12	205.5100	298.3400	233.7600	284.0100	244.6800	1
13	204.9900	298.6200	236.1600	286.2000	243.7300	1
14	204.9500	294.3900	232.0300	284.7500	243.2700	1
15	207.8600	296.2200	234.3900	282.5600	240.3100	1
16	204.2500	294.9100	228.1500	286.5600	246.7600	1
17	209.8800	298.6000	237.2600	285.8000	244.3100	1
18	206.4900	296.6700	229.8100	284.7800	242.5000	1
19	208.8200	298.8400	235.9600	286.9100	240.5500	1
20	206.2500	298.8400	235.8800	286.2200	240.6900	1

Fig. 14.- External dataset example. Main window.

The PCA model can be computed using the “Phase I >> PCA >> Compute PCA” menu option. When using simulations, the data is obtained under Normal Operating Conditions, because of the values selected for feed changes parameters (see section 3.1.2). When using external data, however, the software has to filter the data in order to get NOC conditions, removing any possible outlier. This is done using the following methodology:

1. Remove observations with SPE or Hotelling’s T^2 statistics greater than 2 times the corresponding 95% UCL. This step removes observations that don´t match the model correlation structure or that represent extreme conditions.
2. From remaining observations, remove the observations with largest values for any of both statistics to have 5% maximum of out-of-control points. This is done because it is expected to have a maximum of 5% of points out of the control limits, when a 0.05 significance level is used to compute those limits (equivalent to 95% confidence level).

In order to apply this methodology, the software first computes a PCA model with the training dataset. Then, if it finds any outlier observations, it shows the SPE and T^2 controls charts, plotting in red those abnormal observations, so the user can see which observations will be removed from training data. In this example, the last observation of the training dataset corresponds to an abnormal situation, as it is showed in the SPE (see Fig. 15 left) and T^2 control charts (see Fig. 15 right). Finally, a second PCA model is built with the remaining dataset, which represents NOC Data. This is the model that will be used in Phase II for model exploitation.

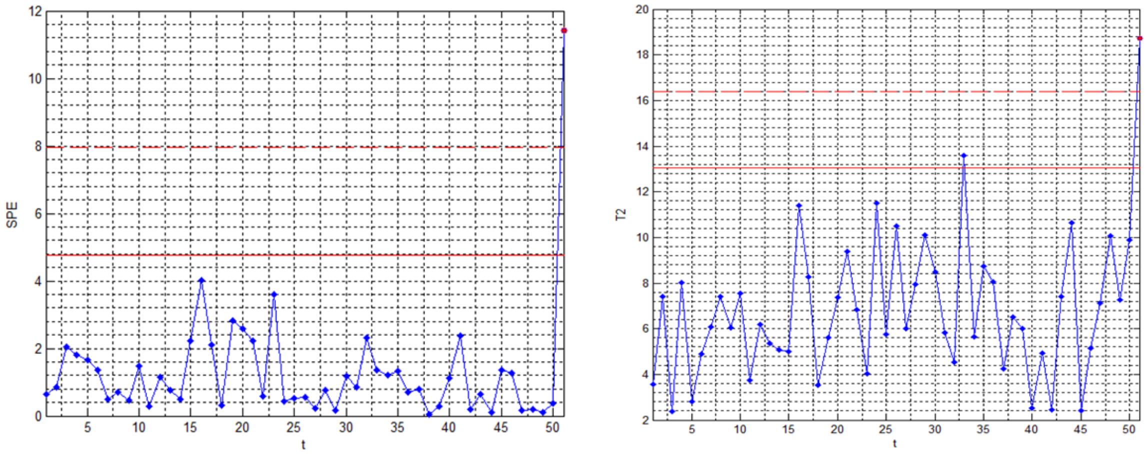


Fig. 15.- SPE and T^2 controls charts for training data.

New data can be added to the project by clicking the “Phase II >> New test” menu option, that will show a dialog window to select the CSV file that contains the data. Using the “Phase II >> PCA >> Model exploitation” we can see that the process is deviating from its Normal Operating Conditions, because the SPE statistic (Fig. 16 left) increases progressively with time and is over its control limits for observation 2 onwards. The contributions for observation #3 (Fig. 16 right) determines that variables z2 and Fi2 are contributing for this abnormal situation.

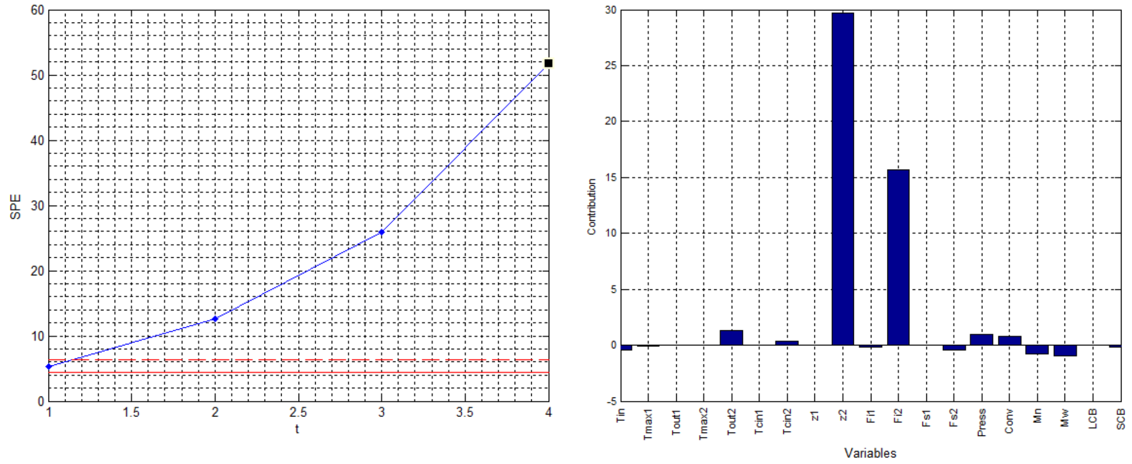


Fig. 16.- SPE chart and contributions of observation #3 for test data.

4.2.2. Pasteurisation process

The second example relates to a pasteurisation process performed in a laboratory-scale pilot plant. The data were collected by an Armfield PCT23 MKII process plant trainer, which permits to monitor the 12 variables listed in Table 11. The instrumentation is equipped with an electrical console for fault simulation.

Table 11.- Variables of the “Pasteurisation” dataset

Variable	Description
LevelT	Feed tank level
T1	Liquid temperature in the pasteurisation tube
T2	Heating water temperature
T3	Final product temperature
T4	Liquid temperature when preheating fresh feed
T5	Fresh feed temperature after preheating
F	Liquid flow rate
P1	Water heating power measure 1
P2	Water heating power measure 2
P3	Water heating power measure 3
Pump1	Feed liquid peristaltic pump opening percentage
Pump2	Heating water peristaltic pump opening percentage

The training set has 3571 observations. The MSPC software excludes 93 observations due to high SPE and/or T^2 values (see Fig. 17), so 3478 observations are considered as NOC data to train the final model, which explains 93.41% of data variability with 8 components (see Fig. 18). The MSPC software selects 8 components automatically based on variance explained (see 3.1.4), but the first two components are enough to explain the simulated scenarios as discussed below, so we select 2 components manually.

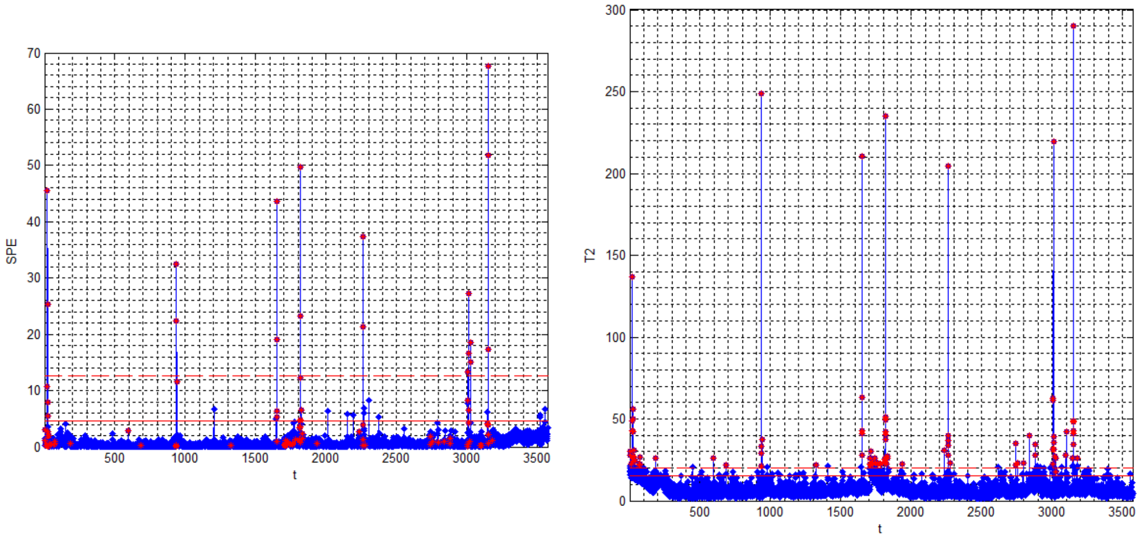
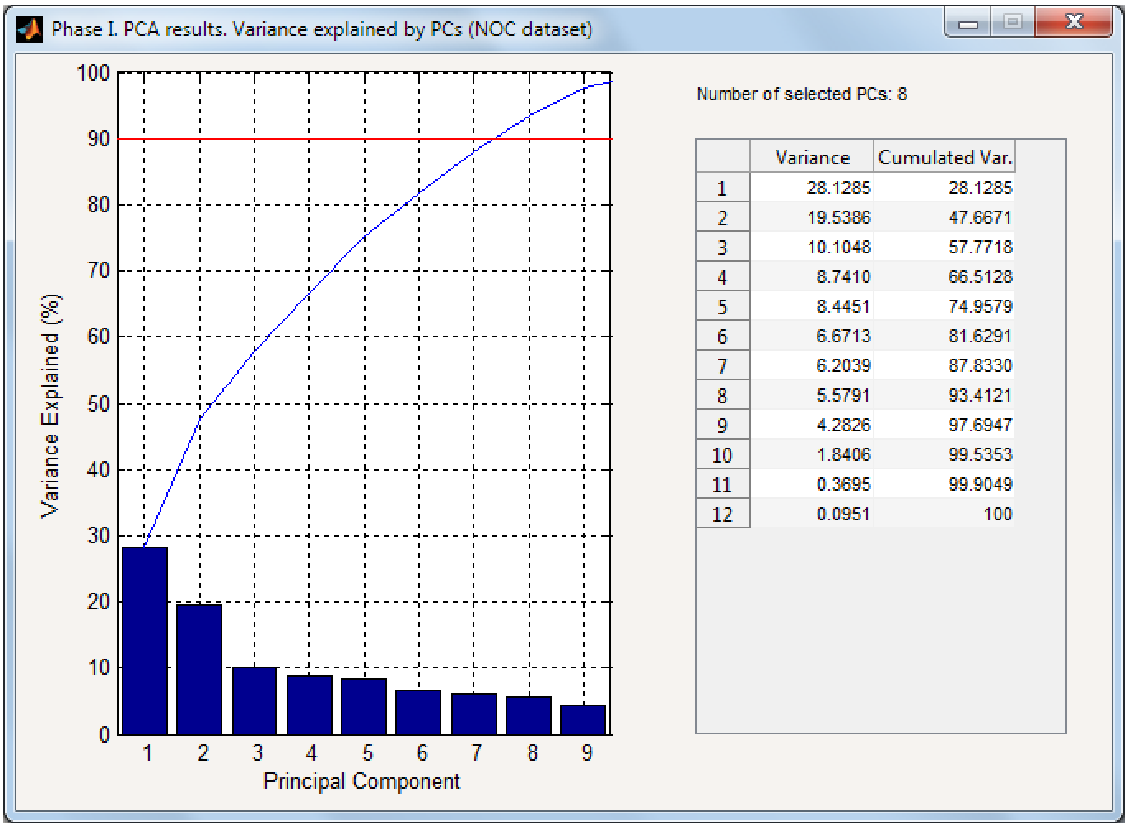


Fig. 17.- SPE and T^2 charts for training data.



Phase I. PCA results. Variance explained by PCs (NOC dataset)

Number of selected PCs: 8

	Variance	Cumulated Var.
1	28.1285	28.1285
2	19.5386	47.6671
3	10.1048	57.7718
4	8.7410	66.5128
5	8.4451	74.9579
6	6.6713	81.6291
7	6.2039	87.8330
8	5.5791	93.4121
9	4.2826	97.6947
10	1.8406	99.5353
11	0.3695	99.9049
12	0.0951	100

Fig. 18.- Variance explained by PCA model using NOC data.

This example simulates two faulty scenarios. The first one corresponds to a flux sensor error. SPE control chart (see Fig. 19) is able to detect this anomaly from observation #34 onwards, and SPE contributions plot signals the variables affected by this fault (“Flow” and “Pump1” variables).

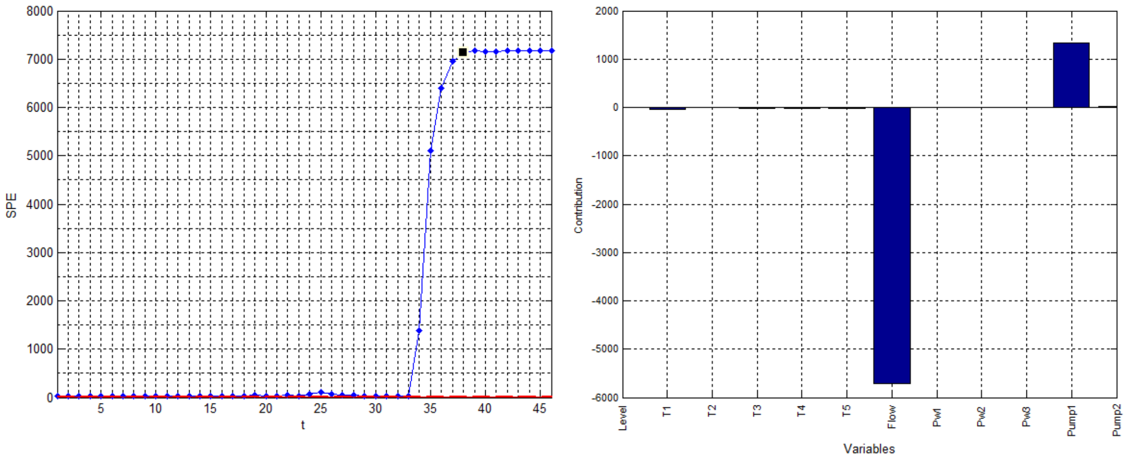


Fig. 19.- SPE chart and contribution plot (flux sensor error, A=2).

The second simulated test corresponds to a T1 sensor fault, as shown in the SPE control chart and its corresponding contribution plot (see Fig. 20).

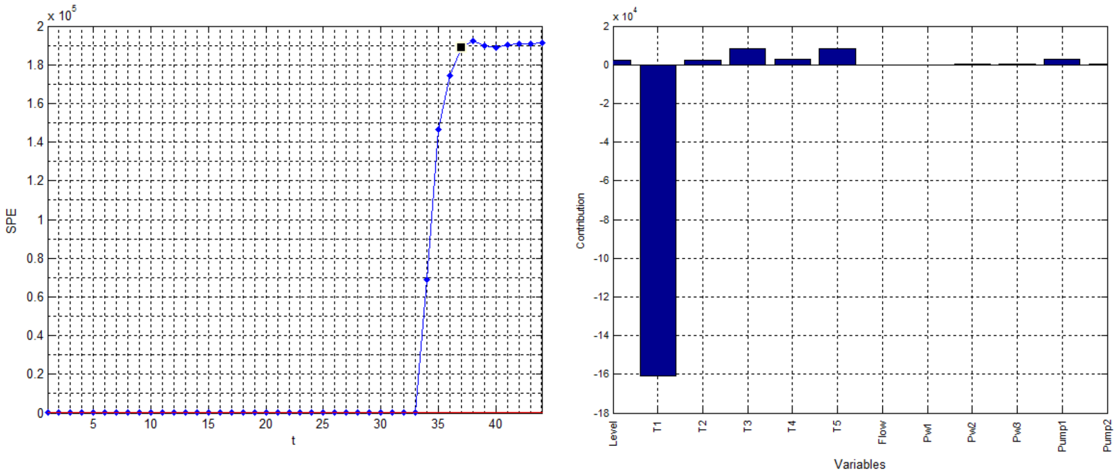


Fig. 20.- SPE chart and contribution plot (feed pump malfunction, A=2).

5. Conclusions

This paper shows a Graphical User Interface (GUI) developed in MATLAB to implement the PCA-based MSPC strategy to a distillation column benchmark. The Simulink model has been fully parameterized in order to create different scenarios. It incorporates the possibility to add noise to the measurements, changes in feed characteristics and operating points, design constraints and several types of disturbances and simulated failures. The user can manipulate all these parameters easily through the implemented GUI.

This software provides an easy way to show the PCA capabilities to monitor multivariate systems and could be used as a simulator for teaching. It also constitutes a good benchmark to generate multivariate datasets (based on the binary distillation column process) that can be exported to Excel files to be analyzed by other tools. The software also incorporates the possibility to analyze external multivariate datasets.

6. Validation

Dr. Marina Cocchi. Associate Professor in Analytical chemistry chemometrics at Dipartimento di Scienze Chimiche e Geologiche, via Campi 103, 41125 Modena (Italy).

We tested the software on Mac OS X 10.11.6 Matlab R2015b and Windows 10, Matlab R2012b and Matlab R2016a. The interface is easy to use and considering the software from a general point of view, the toolbox offers a neat and comprehensive way to exploit the potentiality of PCA for the analysis of Process Data, providing all the plots and features, which are necessary for an effective Multivariate Statistical Process Control framework implementation.

Dr. Carl Duchesne. Professor at Département de génie chimique, Université Laval, Québec G1V 0A6, Canada.

I have successfully installed the software on MATLAB version 8.6.0.267246 (R2015b). Installation was straightforward. I was able to reproduce all the simulations proposed in the paper and I got very similar results to those shown in the Figures (differences due to different noise realizations). The PCA results in both phase I and II (for all tests listed in Table 9) were also very close those reported in the paper. In some cases, the signs of the scores were flipped, but this situation is common (and expected). The software tool is simple and easy to use. I also found it is quite robust. I tried a few additional simulations to those suggested in the paper and it worked well. It has never crashed.

Other than that, I found the auto model selection for PCA (selection of NOC points) interesting. The distillation tower simulations are very relevant for chemical engineering students. I definitely will use the software in the latent variable courses I am preparing now.

Conflict of interests

There is no conflict of interest.

Acknowledgments

Research in this study was partially supported by the Spanish Ministry of Science and Innovation and FEDER funds from the European Union through grant DPI2014-55276-C5-1R

Indication of figures and tables

References

1. MacGregor, J.F., Kourti, T., 1995. Statistical process control of multivariate processes. Control Eng. Pract. 3, 403–414.

2. Ferrer, A., 2007. Multivariate Statistical Process Control Based on Principal Component Analysis (MSPC-PCA): Some Reflections and a Case Study in an Autobody Assembly Process. Qual. Eng. 19, 311–325.

3. Jackson, J. E., 2003. A User´s Guide to Principal Components, Wiley, New York.

4. Kourti, T., 2005. Application of latent variable methods to process control and multivariate statistical process control in industry. Int. J. Adapt. Control Signal Process. 19, 213–246.

5. Martens, H., Naes, T., 1989. Multivariate Calibration, Wiley, New York.

6. Skogestad, S., 1997. Dynamics and control of distillation columns: A tutorial introduction. Chem. Eng. Res. Des. Trans. Inst. Chem. Eng. Part A 75, 539–562.

7. Skogestad, S. Distillation: Column A. http://www.nt.ntnu.no/users/skoge/book/matlab_m/cola/cola.html (accessed 09.07.2014).

8. Woodall, W.H., 2000. Controversies and contradictions in statistical process control. J. Qual. Technol. 32, 341–350.

9. Wold, S., Esbensen, K., Geladi, P., 1987. Principal component analysis. Chemom. Intell. Lab. Syst. 2, 37–52.

10. Wold, S., 1978. Cross-Validatory Estimation of the Number of Components in Factor and Principal Components Model. Technometrics 20, 397–405.

11. Camacho, J., Ferrer, A., 2012. Cross-validation in PCA models with the element wise k-fold (ekf) algorithm: theoretical aspects. J. Chemom. 26, 361–373.

12. Eriksson, L., Johansson, E., Kettaneh-Wold, N., Wold, S., 2001. Multi- and Megavariate Data Analysis: Principles and Applications, Umetrics AB.

13. Tracy, N.D., Young, J.C., Mason, R.L., 1992. Multivariate Control Charts for Individual Observations. J. Qual. Technol. 24, 88–95.

14. Kourti, T., MacGregor, J.F., 1996. Multivariate SPC Methods for Process and Product Monitoring. J. Qual. Technol. 28, 409–428.

15. Box, G.E.P., 1954. Some Theorems on Quadratic Forms Applied in the Study of Analysis of Variance Problems, I. Effect of Inequality of Variance in the One-Way Classification. Ann. Math. Stat. 25, 290–302.

16. Nomikos, P., MacGregor, J.F., 1995. Multivariate SPC Charts for Monitoring Batch Processes. Technometrics 37, 41–59.

17. Sales, R.F., Vitale, R., de Lima, S.M., Pimentel, M.F., Stragevitch, L., Ferrer, A., 2016. Multivariate statistical process control charts for batch monitoring of transesterification reactions for biodiesel production based on near-infrared spectroscopy. Comput. Chem. Eng. 94, 343–353.

18. [dataset] Dunn, K., 2011. Low-density polyethylene (LDPE) production process dataset. http://openmv.net/info/ldpe (accessed 30.03.2017).

19. MacGregor, J.F., Jaeckle, C., Kiparissides, C., Koutoudi, M., 1994. Process monitoring and diagnosis by multiblock PLS methods. AIChE J. 40, 826–838.

S-1. Simulink model

On the other hand, the Simulink model is fully parameterized to ease the creation of different scenarios, such as feed disturbances, regulatory control failures and transitions between operating points.

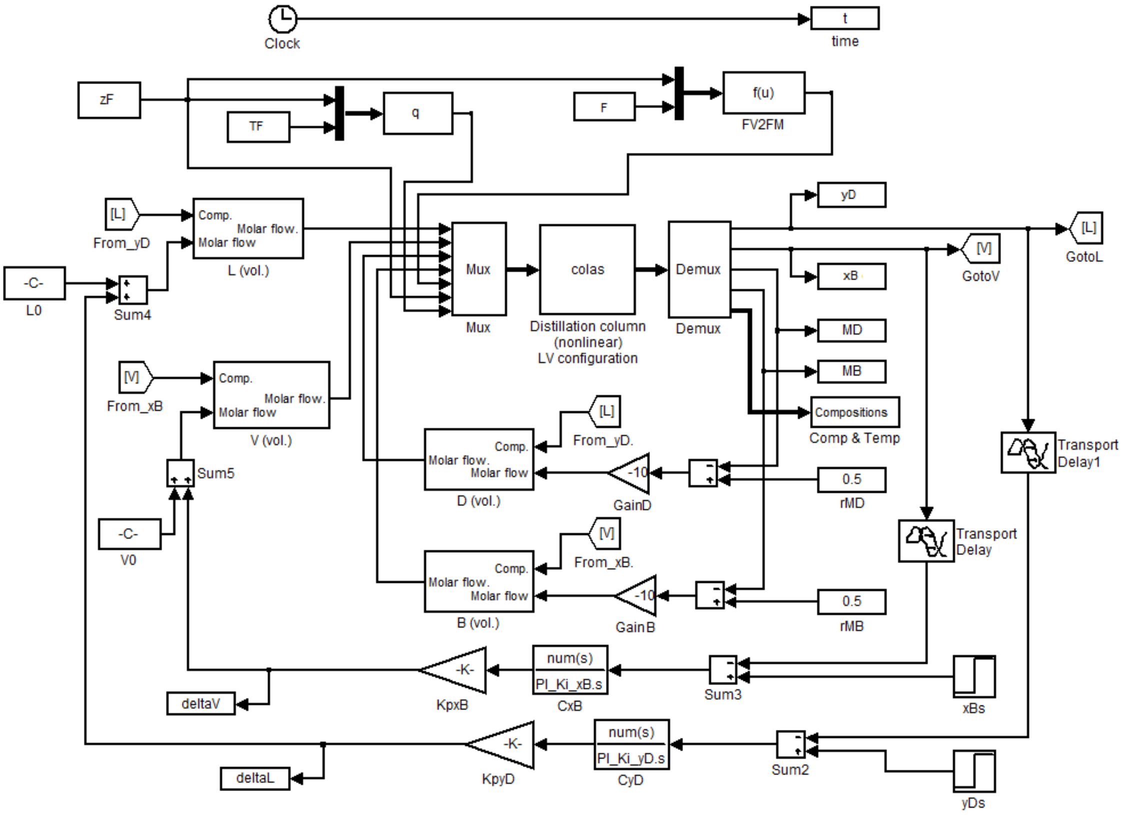


Fig. S-21.- Nonlinear distillation column Simulink model.

The “colas” block in the center of the Simulink model is an S-Function block with the system of differential equations derived from mass and energy balances in the distillation column. It has 7 inputs:

- *L, V, D, B*: molar flows (mol/min)
- *FM*: feed molar flow (mol/min)
- z_F: feed composition
- *q*: feed quality factor

The Skogestad’s original model uses molar flows and the feed quality factor (*q*). Measurements in a typical industrial plant are volumetric flows and temperatures. So, in order to get an industrial-like model, molar flows are converted to volumetric flows (see “L (Vol.)”, “V (Vol.)”, “D (Vol.)”, “B (Vol.)” blocks in Fig. 2). These blocks use compositions to compute volumetric flows from molar flows. Something equivalent is done for feed composition: feed volumetric flow *F* (L/h) is converted to *FM* (feed molar flow, mol/min) using feed composition z_F (see block “FV2FM” in Fig. 2). On the other hand, an S-Function is added (see block “q” in Fig. 2) to compute feed quality factor q from feed composition (z_F) and feed temperature (T_F). Now we have the following variables:

- L, V, D, B: volumetric flows (L/h)
- F: feed volumetric flow (L/h)
- T_F: feed temperature (ºC)

The outputs of the S-Function block “colas” are the holdups in the reboiler (see “MB” block in Fig. 2) and in the separator drum (see “MD” block in Fig. 2), and the compositions and temperatures at the N_S stages in the distillation column:

- Stage 1: refers to bottom.
- Stage 2-40: refers to column trays 1 to 39
- Stage 41: refers to distillate

The simulation computes M_D and M_B in kmol. In practice, this kind of information is not available but could be replaced with a fluid level measurement. It is possible to convert these holdups to fluid level taking into account compositions and densities but this conversion has been discarded.

As stated before, this model uses LV-configuration for control purposes, so it assumes that bottom composition (x_B) and distillate composition (y_D) are known. In practice, in most cases, compositions are not available and, if they are, measurements come from sporadic laboratory analysis and on-line measurements are seldom available. When compositions are not available, it is also possible to use bottom temperature (T_1) and distillate temperature (T_{41}) instead of the corresponding compositions (x_B and y_D) for control purposes. In the case of feed composition (z_F), if it is not available it will be considered as an unmeasured disturbance.

The Simulink model in Fig. 2 also shows the aforementioned control loops:

- The deviation of distillate composition (y_D) from set point (“yDs” block) is used to compute the control action (“deltaL” block) of the PI controller, which is a variation regarding the initial conditions (“L0” block). This computes a new top internal flow (L) for the system in order to keep distillate composition inside specifications. Something equivalent is done for bottom internal flow (V) and bottom composition (x_B) in the other PI control loop.
- The deviation of distillate holdup (“MD” block) from set point (“rMD” block) is used to compute a new distillate flow (D) to control the liquid level in the separator drum in the top of the column (distillate holdup level M_D). Something equivalent is done for bottom flow (B) and bottom holdup level (“MB” block).

S-2. Simulated data

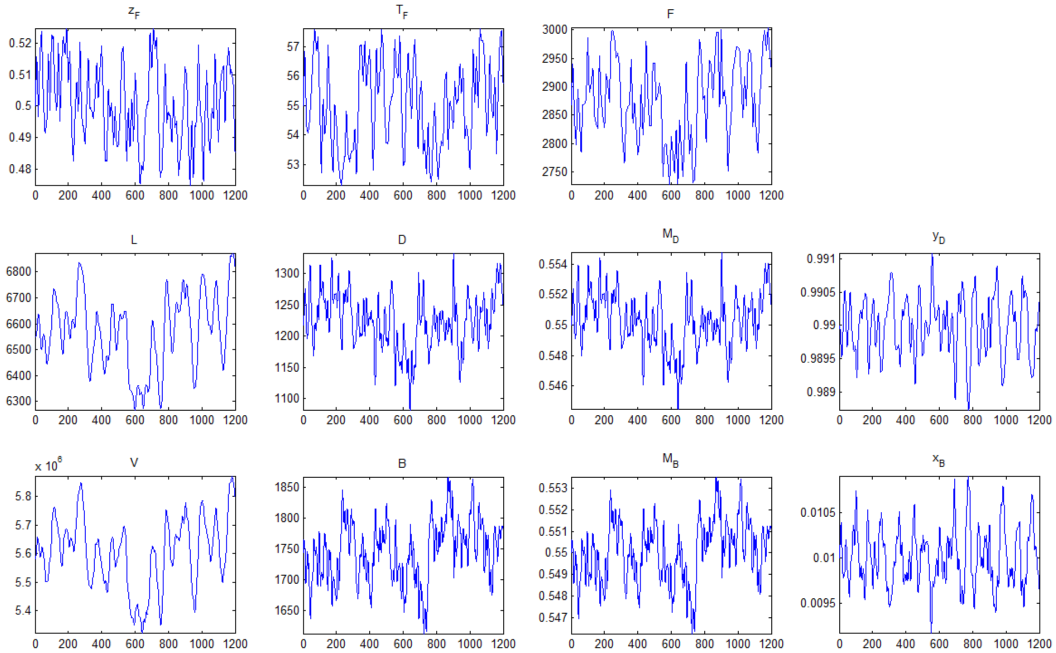


Fig. S-22.- Phase I simulation results.

S-3. Benchmark simulations

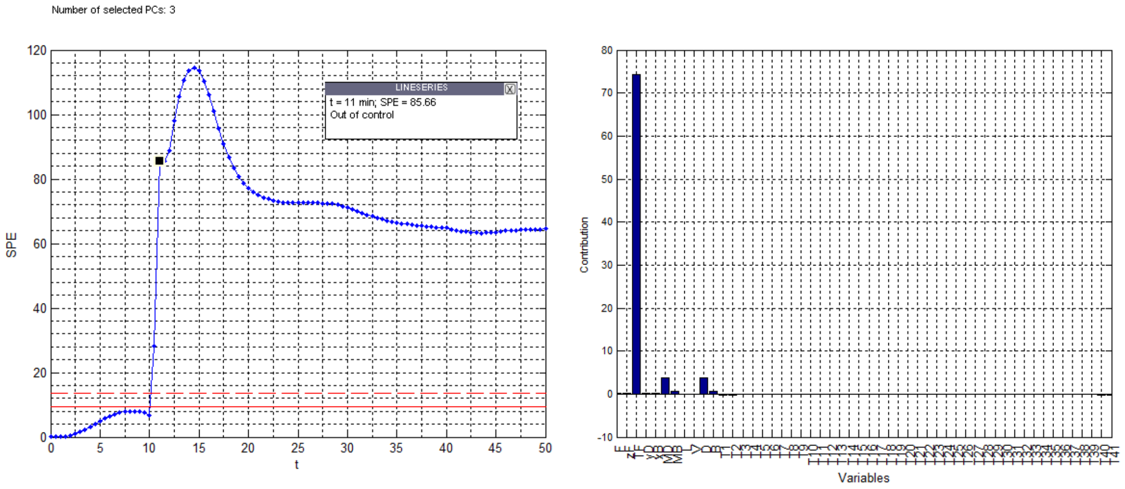


Fig. S-23.- Test#5 (T-ramp). SPE chart (left) and contribution plot (right).

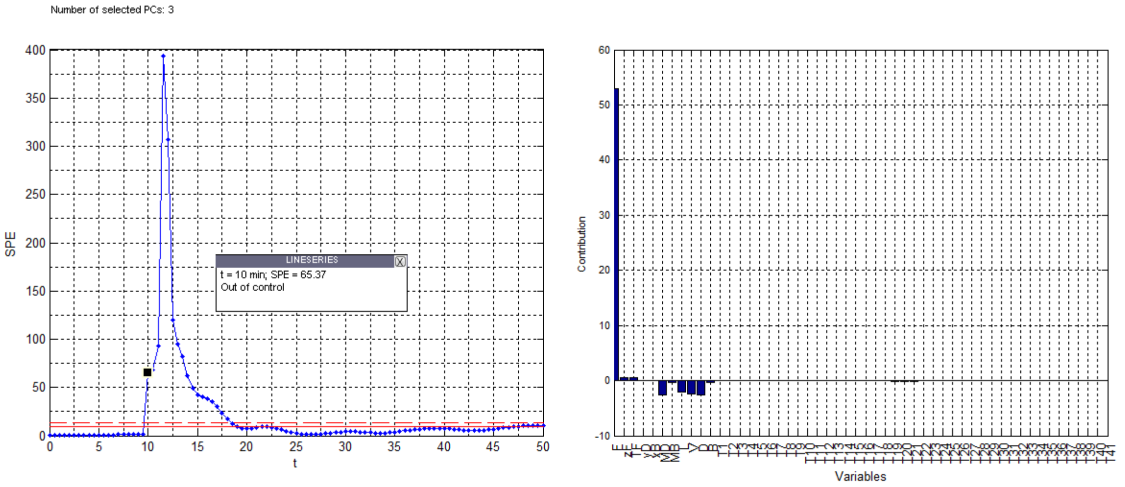


Fig. S-24.- Test#9 (F-pulse). SPE chart (left) and contribution plot (right).

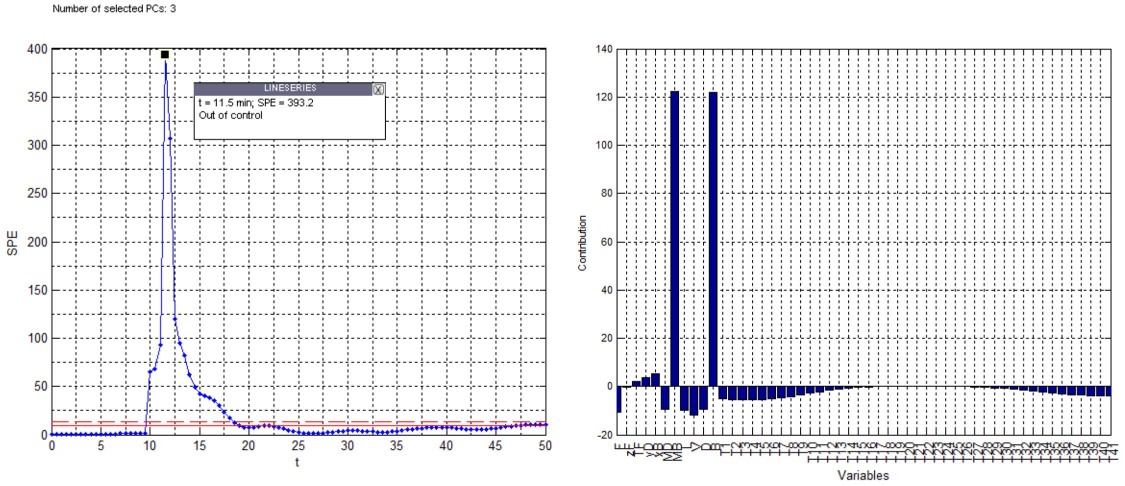


Fig. S-25.- Test#9. SPE chart (left) and contribution plot (right). Effect in bottom holdup M_B.

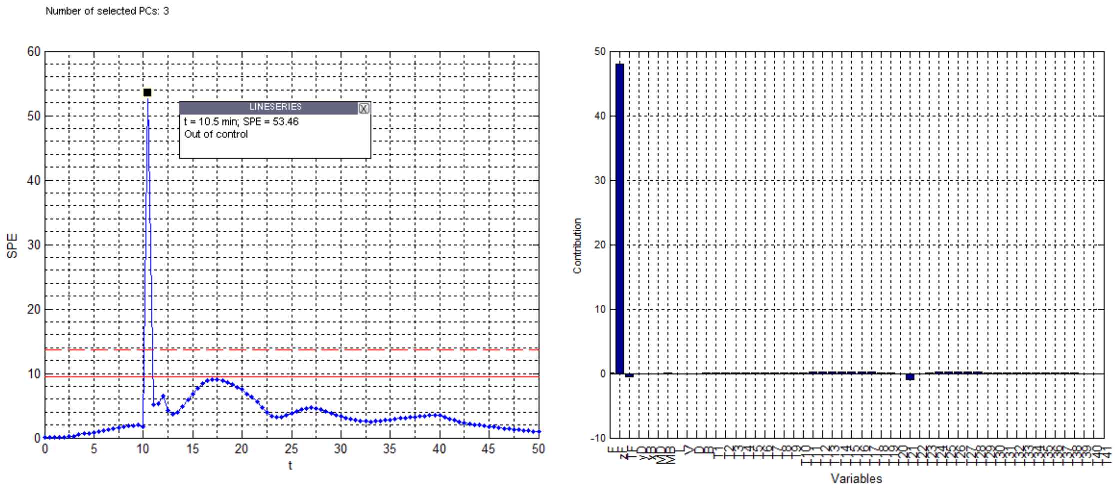


Fig. S-26.- Test#1 with a 15% disturbance size. SPE chart (left) and contribution plot (right).

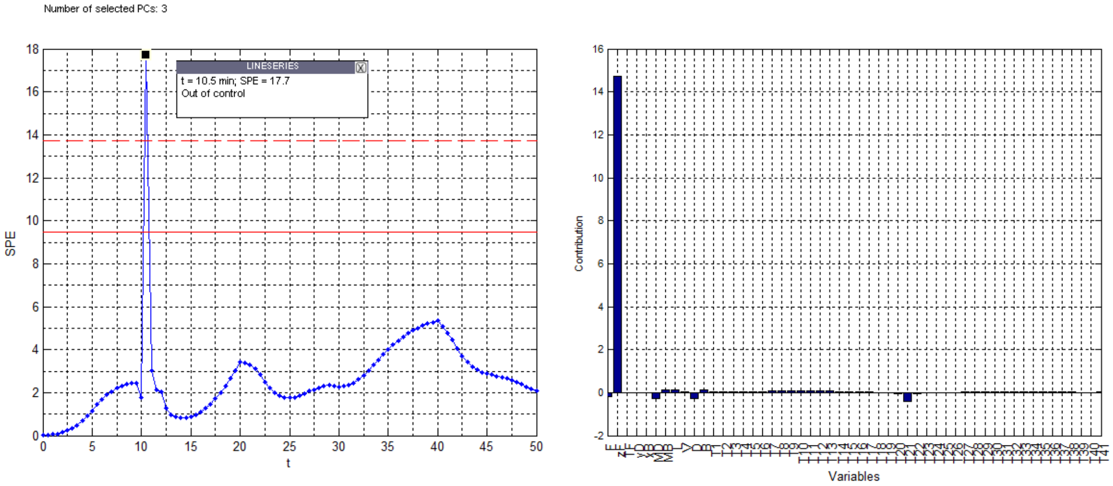


Fig. S-27.- Test#1 with a 10% disturbance size. SPE chart (left) and contribution plot (right).

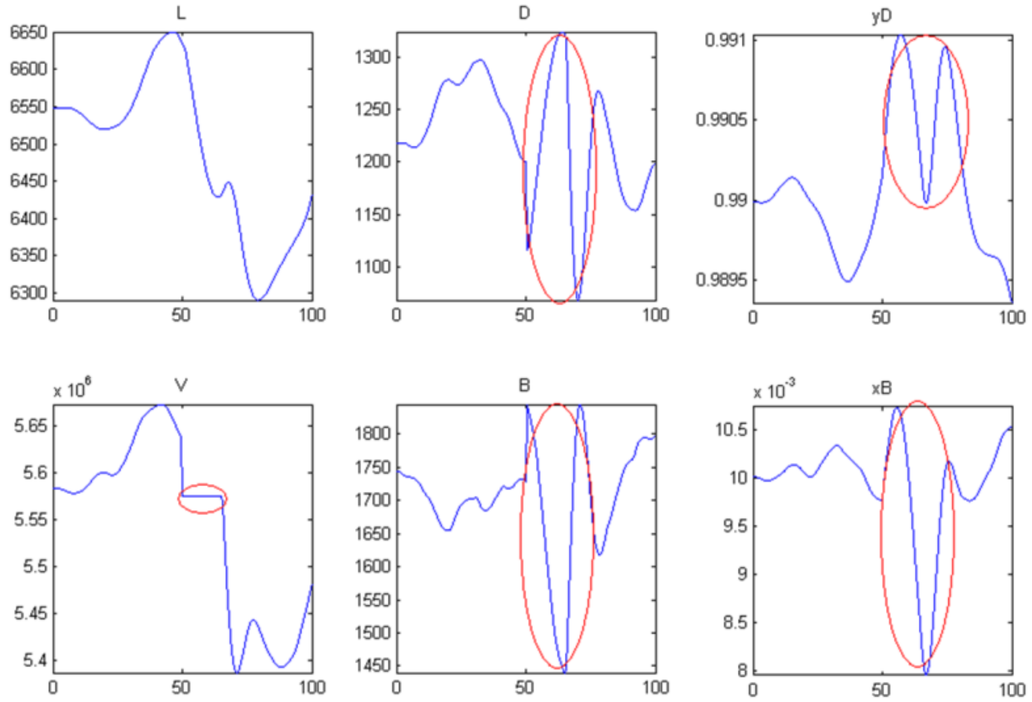


Fig. S-28.- Several variables plots for test#10 (PI failure x$_B$)

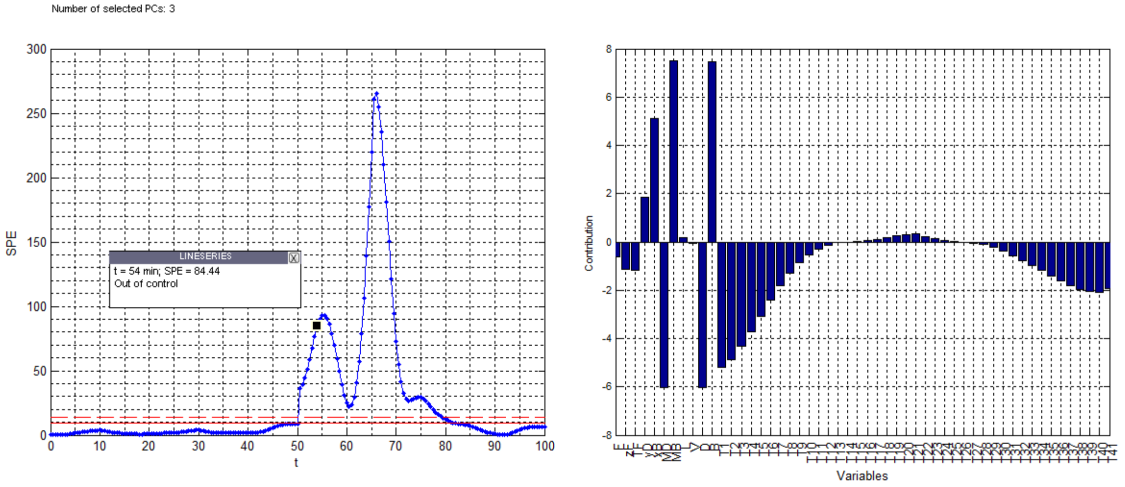


Fig. S-29.- Test#10 (PI failure x_B). SPE chart (left) and contribution plot (right).

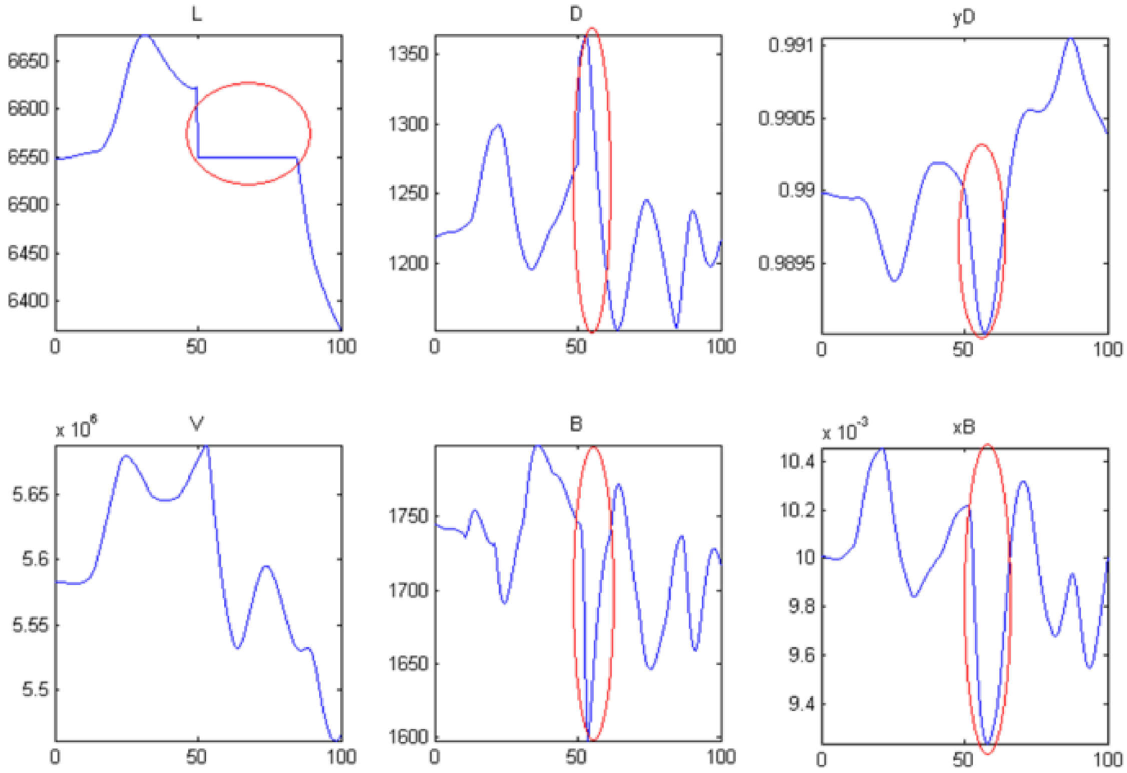


Fig. S-30.- Several variables plots for test#11 (PI failure y_D)

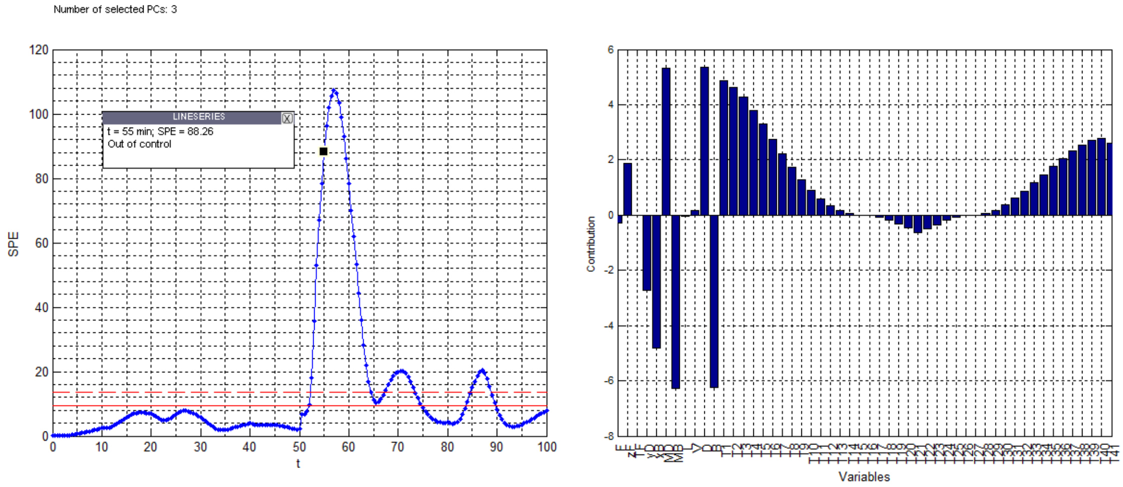


Fig. S-31.- Test#11 (PI failure y_D). SPE chart (left) and contribution plot (right).

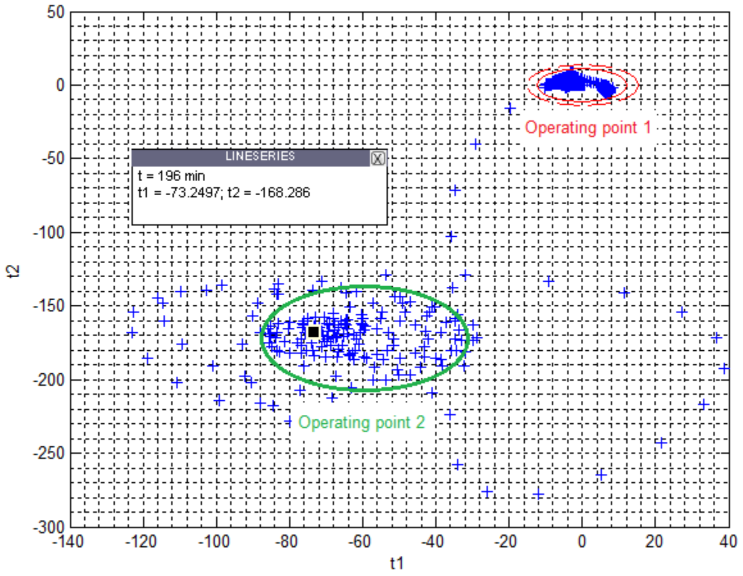


Fig. S-32.- Scatter plot for the first two components for test#12

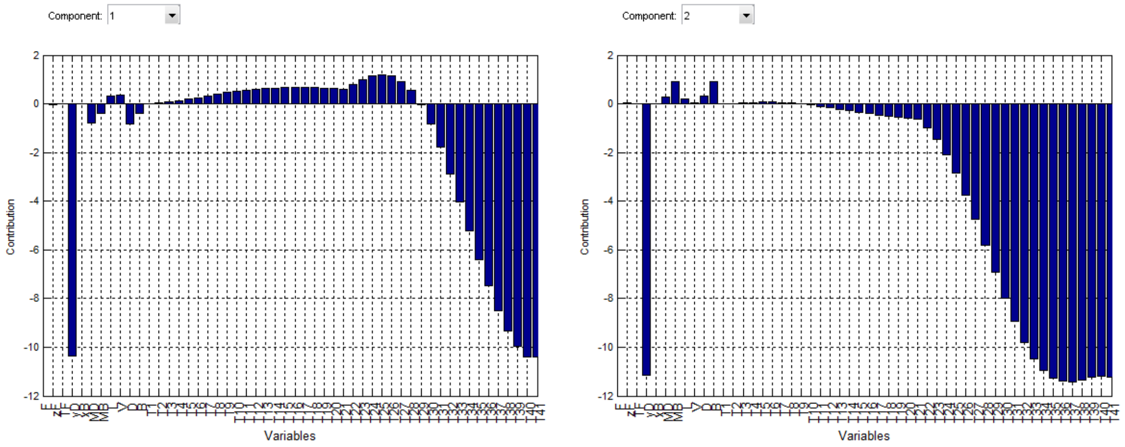


Fig. S-33.- Contribution plots for the first two components (test#12)

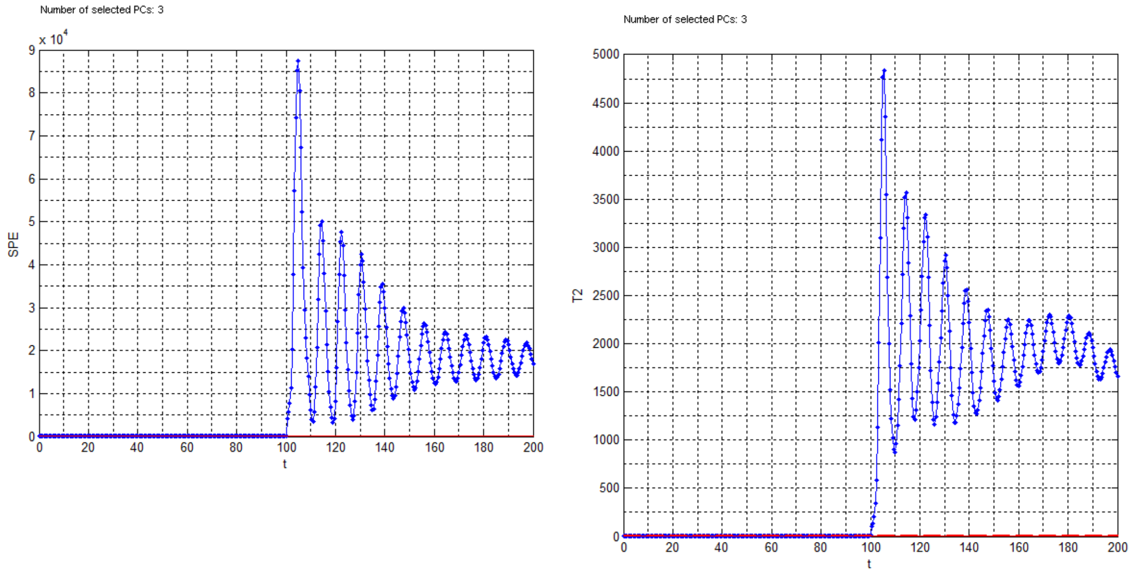


Fig. S-34.- Test#12. SPE (left) and T^2 (right) control charts.

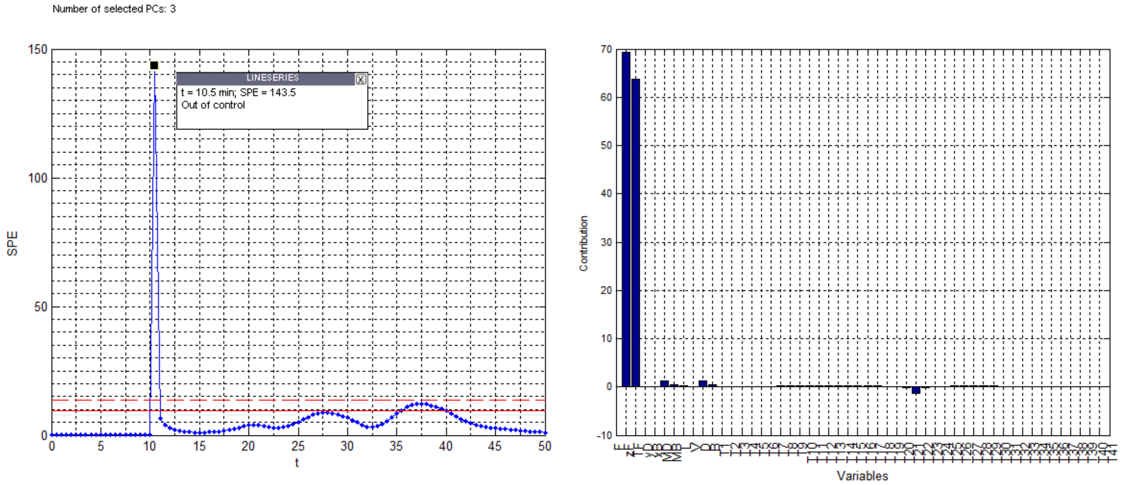


Fig. S-35.- Test#14 (z&T pulse). SPE chart (left) and contribution plot (right).

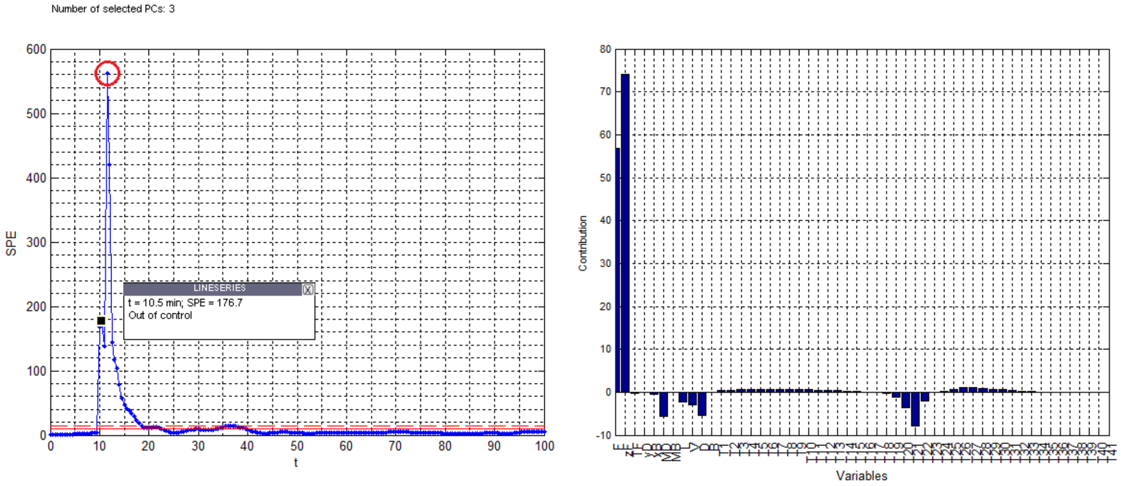


Fig. S-36.- Test #15 (F&z pulse). SPE chart (left) and contribution plot (right).

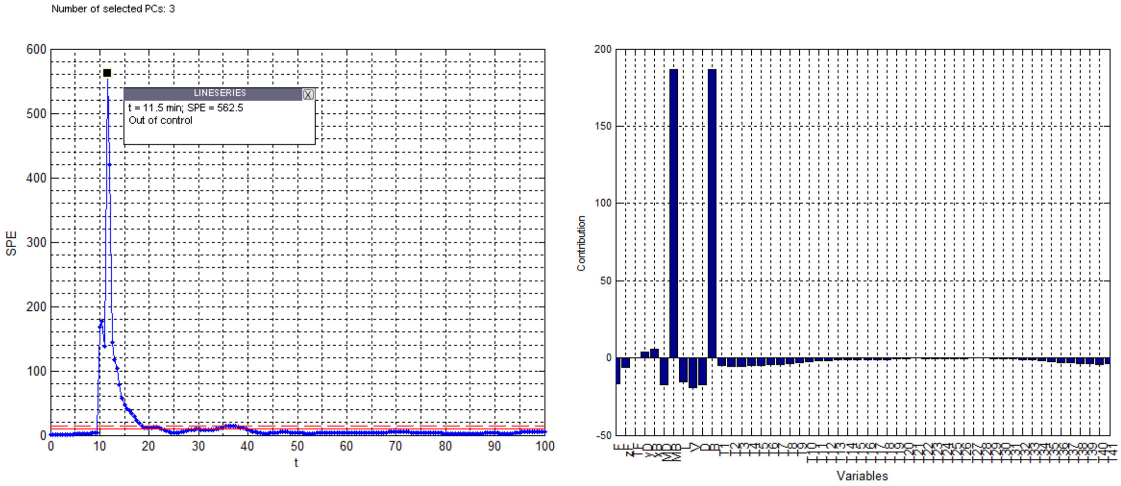


Fig. S-37.- Test #15 (F&z pulse). SPE chart (left) and contribution plot (right). Effect in variable *B*.