Master thesis

Multivariate statistical process monitoring of a distillation column

Dr. Pedro Miguel Villalba Torán

Director: Dr. Alberto-José Ferrer Riquelme

Codirector: Dr. Javier Sanchis Saez

Valencia, September 2012

Departamento de Estadística, Investigación Operativa Aplicadas y Calidad
Máster en Ingeniería de Análisis de Datos, Mejora de Procesos y Toma de Decisiones

CONTENTS

1. Objectives

The main objective of this thesis is to create a benchmark for PI tuning optimization and multivariate statistical process monitoring from a Simulink nonlinear model of a binary distillation column developed by Skogestad [1]. To achieve this, the following objectives are proposed:

1. Modify the original model developed by Skogestad [1] to obtain a more realistic one.
2. Study optimization techniques for PI tuning.
3. Study types of disturbances in distillation columns.
4. Study a methodology for PCA monitoring.

2. Distillation

2.1. Introduction

Distillation is a process of physically separating a mixture into two or more products that have different boiling points, by preferentially boiling the more volatile components out of the mixture. When a liquid mixture of two volatile materials is heated, the vapor that comes off will have a higher concentration of the more volatile (*i*.e., lower boiling-point) material than the liquid from which it was evolved. Conversely, if a vapor is cooled, the less volatile (*i*.e., higher boiling-point) material has a tendency to condense in a greater proportion than the more volatile material.

2.2. Fractionation design

2.2.1. Introduction

Continuous rectification, or fractionation, is a multistage countercurrent distillation operation. For a binary solution, with certain exceptions, it is ordinarily possible by this method to separate the solution into its components, recovering each in any state of purity desired.

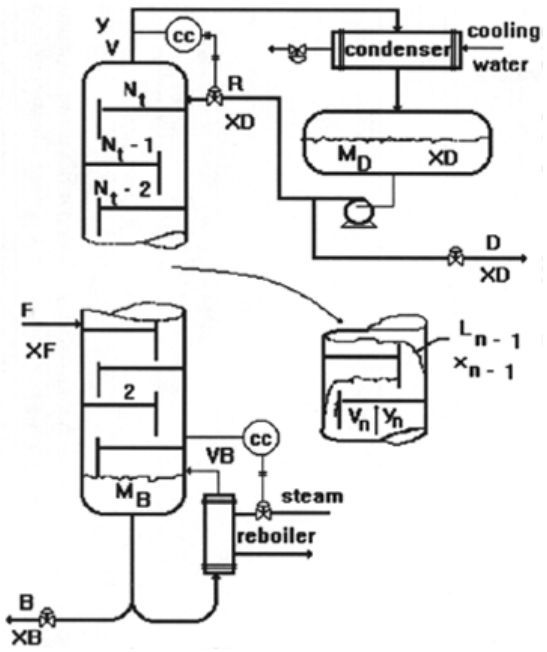


Figure 2.1.- Fractionation scheme

Notation

L_i and V_i: liquid and vapor flow from stage i (kmol/min)

x_i and y_i: liquid and vapor composition of light component on stage i (mole fraction)

M_i: liquid holdup on stage i (kmol)

D and B: distillate (top) and bottoms product flow rate (kmol/min)

L=L_T and V=V_B: reflux flow and boilup flow (kmol/min)

F, z_F: Feed rate (kmol/min) and feed composition (mole fraction)

q_F: fraction of liquid in feed

i: stage no. (1=bottom. NF=feed stage, NT=total condenser)

The feed is introduced more or less centrally into a vertical cascade of stages. Vapor rising in the section above the feed (called the absorption, enriching, or rectifying section) is washed with

liquid to remove or absorb the less volatile component. Since no extraneous material is added, as in the case of absorption, the washing liquid in this case is provided by condensing the vapor issuing from the top, which is rich in more volatile component. The liquid returned to the top of the tower is called reflux, and the material permanently removed is the distillate, which may be a vapor or a liquid, rich in more volatile component. In the section below the feed (stripping or exhausting section), the liquid is stripped of volatile component by vapor produced at the bottom by partial vaporization of the bottom liquid in the reboiler. The liquid removed rich in less volatile component, is the residue, or bottoms. Inside the tower, the liquids and vapors are always at their bubble points and dew points, respectively, so that the highest temperatures are at the bottom, the lowest at the top. The entire device is called a fractionator.

2.2.2. Model assumptions

The model considered in this work corresponds to a fractionator of a mixture of two components (methanol and ethanol) operating at constant pressure. Although the objective is to obtain a rigorous model, we can make two simplifying assumptions [(4)]:

1. Constant relative volatility. In this case the vapor-liquid equilibrium between any two components is given by Eq. 10.1, where α is independent of composition (and usually also of pressure). This assumption holds well for the separation of similar components, for example, for alcohols or for hydrocarbons.

2. Constant molar flows. In this case the molar flows of liquid and vapor along the column do not change from one stage to the next, that is, if there is no feed or product removal between stages *i* and *i+1*, then, at steady state:

$$L_i = L_{i+1} \; ; \; V_i = V_{i+1}$$

   Again, this assumption usually holds well for similar components if their heats of vaporization do not differ too much, which is the case of the two selected components (methanol and ethanol).

2.2.3. Design parameters

The column proposed in this work is designed to produce a distillate product with D = 0.5 kmol/min and composition y_D = 0.99 and a bottoms product with B = 0.5 kmol/min and composition x_B = 0.01, from a F = 1 kmol/min equimolal feed of methanol/ethanol at 55ºC. So we have the following nominal conditions:

Feed rate F = 1 (kmol/min)
Feed composition z_F = 0.5 (mole fraction units)
Feed temperature T_F = 55ºC
y_D = 0.99, D = 0.55 kmol/min
x_B = 0.01, B = 0.55 kmol/min

With z_F = 0.5 and T_F = 55ºC we calculate the q factor of the feed: q = 1.0405 (see example in section 2.2.4).

The reflux flow is L = 2.6889 kmol/min and the boilup flow is V = 3.2294 kmol/min.

The nominal liquid holdup is M_i = 0.5 kmol for all stages, including reboiler and condenser.

The relative volatility of methanol-ethanol system is considered to be 1.5

For this configuration, if we apply the McCabe-Thiele method (see Appendix II - McCabe-Thiele method) we find that the column has NT=41 stages including reboiler and total condenser and the feed is at stage NF = 21 counted from the bottom.

2.2.4. Feed quality

The original model created by Skogestad [1] works with the parameter q that is a measure of feed quality that derives from the McCabe-Thiele method for column design. As stated above, one of the objectives of this work is to obtain a model as realistic as possible, so we will change q factor with other variables more conventional in industrial environments. The methodology presented in Appendix III - Feed quality computes q factor from feed composition (z) and feed temperature (T).

This change lets the model be more flexible, easing the connection with other Simulink blocks (see section 9.1.1).

2.2.5. Operating points

Process industries usually work with different qualities in their products. In this case, we will suppose that the distillation column must obtain three different qualities, that is, three different operating points:

Table 2.1.- Operating points

Operating point	x_B	y_D	D (kmol/min)	B (kmol/min)	L (kmol/min)	V (kmol/min)
1	0.01	0.99	0.55	0.55	2.6889	3.2294
2	0.01	0.96	0.55	0.55	2.2537	2.8100
3	0.05	0.99	0.55	0.55	2.3243	2.8435

The nominal conditions for the feed will always be the same (see section 2.2.3): z = 0.5, T = 55ºC and F = 1 kmol/min.

We must remember that compositions are given for the most volatile component, *i*.e. methanol. So the first operating point produces 0.55 kmol/min distillate of 99% purity in methanol and a bottoms product of 0.55 kmol/min of 99% purity in ethanol ($x_{ethanol}$ = 1-$x_{methanol}$ = 1-0.01 = 0.99).

3. Simulation

3.1. Model equations

In previous section, we have used the McCabe-Thiele method to design the distillation column. This method is based on energy and mass balances for the column in steady state. Now, we will develop these balances in differential form to simulate the behavior of the column when feed changes.

The following model equations apply at all stages except in the top (condenser), feed stage and bottom (reboiler) [3]

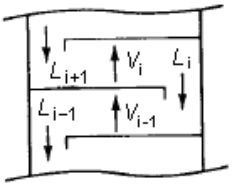


Figure 3.1.- *Ith* stage [3]

1. Total material balance on stage *i*:

$$\frac{dM_i}{dt} = L_{i+1} - L_i + V_{i-1} - V \qquad \textbf{Eq. 3.1}$$

2. Material balance for light component on each stage *i*:

$$\frac{d(M_i x_i)}{dt} = L_{i+1}x_{i+1} - L_i x_i + V_{i-1} y_{i-1} - V_i y \qquad \textbf{Eq. 3.2}$$

that gives the following expression for the derivative of the liquid mole fraction:

$$\frac{dx_i}{dt} = \frac{1}{M_i}\left(\frac{d(M_i x_i)}{dt} - x_i \frac{dM_i}{dt}\right) \qquad \textbf{Eq. 3.3}$$

3. Vapor-liquid equilibrium (vapor phase). The vapor composition y_i is related to the liquid composition x_i on the same stage through the algebraic vapor-liquid equilibrium:

$$y_i = \frac{\alpha x_i}{1 + (\alpha - 1)x_i} \qquad \textbf{Eq. 3.4}$$

where α is the relative volatility (see Appendix I - Relative volatility). From the assumption of constant molar flows and no vapor dynamics we have the following expression for the vapor flows:

$$V_i = V_{i-1}$$ **Eq. 3.5**

except at the feed stage, where:

$$V_{NF} = V_{NF-1} + (1 - q_F) F$$ **Eq. 3.6**

4. Vapor-liquid equilibrium (liquid phase). The liquid flows depend on the liquid holdup on the stage above and the vapor flow as follows (this is a linearized relationship):

$$L_i = L_{0i} + \frac{M_i - M_{0i}}{\tau} + (V - V_0)_{i-1} \lambda$$ **Eq. 3.7**

where L_{0i} (kmol/min) and M_{0i} (kmol) are the nominal values for the liquid flow and holdup on stage *i*. The vapor flow into the stage may also affect the holdup; lambda may be positive because more vapor may give more bubbles and thus may push liquid off the stage. If lambda is large (larger than 0.5) then the reboiler holdup "flattens out" for some time in response to an increase in boilup, and if lambda > 1 we get an inverse response. Lambda may also be negative if the increased pressure drop caused by larger *V* results in a larger holdup in the downcomers. In general it is difficult to estimate lambda for tray columns. For packed columns lambda is usually close to zero.

The parameter τ is the time constant (min) for liquid flow dynamics on each stage.

The above equations apply at all stages except in the top (condenser), feed stage and bottom (reboiler):

Feed stage, *i* = *F* (we assume the feed is mixed directly into the liquid at the feed stage):

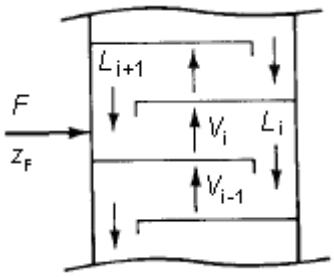


Figure 3.2.- Feed stage [3]

$$\frac{dM_i}{dt} = L_{i+1} - L_i + V_{i-1} - V_i + F$$ **Eq. 3.8**

$$\frac{d(M_i x_i)}{dt} = L_{i+1} x_{i+1} - L_i x_i + V_{i-1} y_{i-1} - V_i y_i + F z_F$$ **Eq. 3.9**

Total condenser, $i = N$ ($M_N = M_D$, $L_N = L$, $y_N = y_D = x_D$):

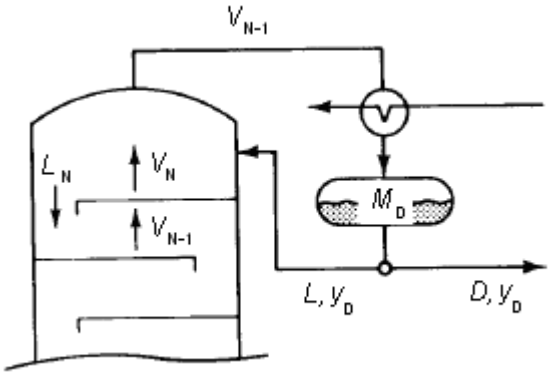


Figure 3.3.- Last stage (condenser) (3)

$$\frac{d(M_i x_i)}{dt} = V_{i-1} y_{i-1} - L y_D - D y_D \qquad \textbf{Eq. 3.10}$$

Reboiler, $I = 1$ ($M_i = M_B$, $V_i = V_B = V$)

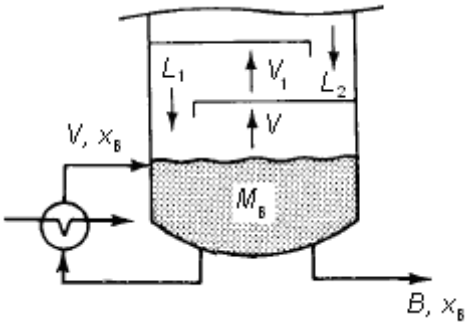


Figure 3.4.- First stage (reboiler) (3)

$$\text{n}\frac{dM_i}{dt} = L_{i+1} - V_i - B \qquad \textbf{Eq. 3.11}$$

$$\frac{d(M_i x_i)}{dt} = L_{i+1} x_{i+1} - V_i y_i - B x_B \qquad \textbf{Eq. 3.12}$$

3.2. Simulink model

3.2.1. Introduction

Simulink is an environment for multidomain simulation and Model-Based Design for dynamic and embedded systems. It provides an interactive graphical environment and a customizable set of block libraries that let you design, simulate, implement, and test a variety of time-varying systems, including communications, controls, signal processing, video processing, and image processing [(2)].

Simulink is integrated with MATLAB, providing immediate access to an extensive range of tools that let you develop algorithms, analyze and visualize simulations, create batch processing scripts, customize the modeling environment, and define signal, parameter, and test data.

Simulink provides a graphical user interface (GUI) for building models as block diagrams, allowing you to draw models as you would with pencil and paper. Simulink also includes a comprehensive block library of sinks, sources, linear and nonlinear components, and connectors. The interactive graphical environment simplifies the modeling process, eliminating the need to formulate differential and difference equations in a language or program.

Models are hierarchical, so you can build them using both top-down and bottom-up approaches. You can view the system at a high level and then double-click blocks to see increasing levels of model detail. This approach provides insight into how a model is organized and how its parts interact.

After you define a model, you can simulate its dynamic behavior using a choice of mathematical integration methods, either from the Simulink menus or by entering commands in the MATLAB Command Window. The menus are convenient for interactive work, while the command line is useful for running a batch of simulations. For example, if you are doing Monte Carlo simulations or want to apply a parameter across a range of values, you can use MATLAB scripts.

Using scopes and other display blocks, you can see the simulation results while the simulation runs. You can then change parameters and see what happens for "what if" exploration. The simulation results can be put in the MATLAB workspace for post processing and visualization.

3.2.2. Distillation column

The original model was created by Skogestad [(1)] (see Figure 3.5). It uses an S-Function block that calls a MATLAB function that implements the system of differential equations shown in section 2.2.3.

This nonlinear model has four manipulated inputs (*L*, *V*, *D* and *B*), three disturbances (*F*, z_F and q_F) and *2N* states returned by S-Function *colas*:

- State 1: liquid composition in reboiler $x_1 = x_B$,
- Then follow the stage compositions x_i up the column, $i = 2, \ldots, N-1$
- State *N*: composition stage *N* (condenser), $x_N = y_D$
- State *N+1*: holdup reboiler, $M_1 = M_B$
- Then follow the stage holdups up the column M_i, $i = N+2, \ldots, 2N-1$
- State *2N*: condenser holdup, $M_{2N} = M_D$

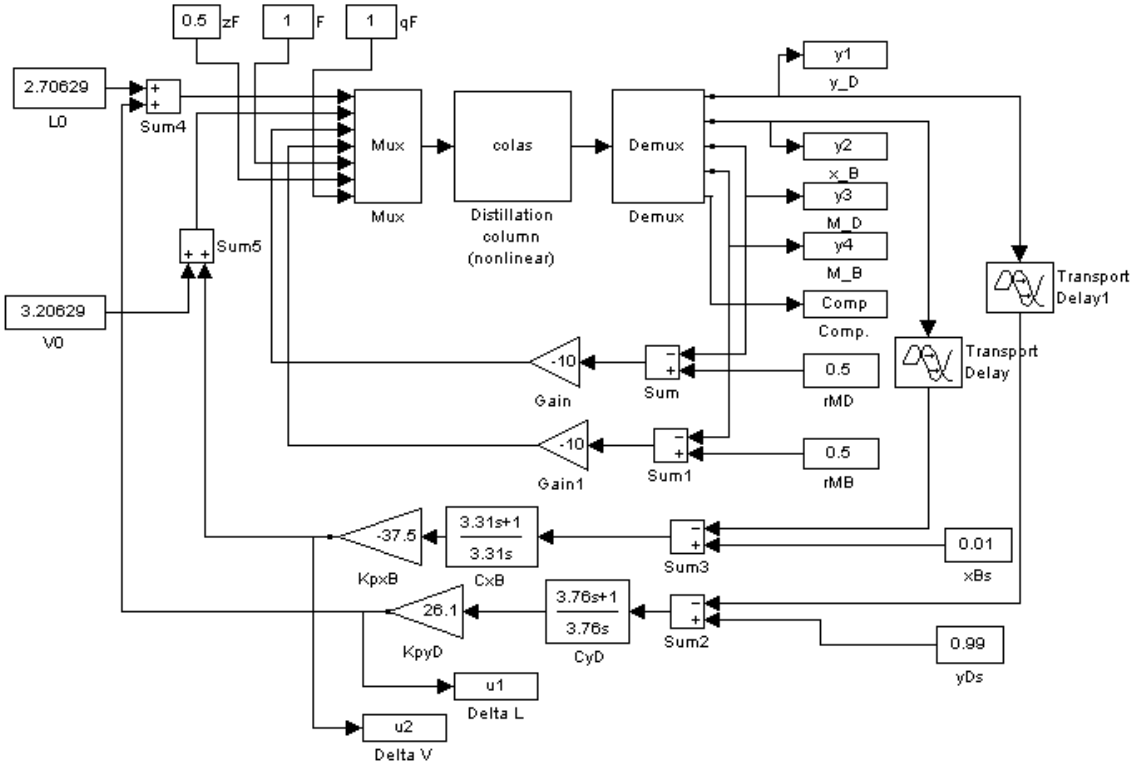


Figure 3.5.- Original Simulink model.

The block parameter “Compositions” (Figure 3.6) contains the compositions of every tray in the column, so plotting their values at a determined time renders the compositions profile of the column at that time. As mentioned above, the first value corresponds to the composition of the bottoms product (x_B) and the last one, to the composition of the distillate (y_D).

The model shown in Figure 3.5 corresponds to a control configuration called LV-configuration. In this case we have two decentralized PI to control y_D and x_B using the manipulated variables *L* and *V*, respectively (see section 4.4 for more details about this configuration). This will be the model used in this work for the study of multivariate statistical process monitoring.

Several improvements have been done to the original model to achieve the objectives proposed in this work that have finally rendered the definitive model shown in Figure 3.6. Those enhancements will be explained in next sections and include the following:

- Model parameterization.
- Addition of noise, disturbances and feed and operating changes.
- Calculation of q factor.
- Design constraints.

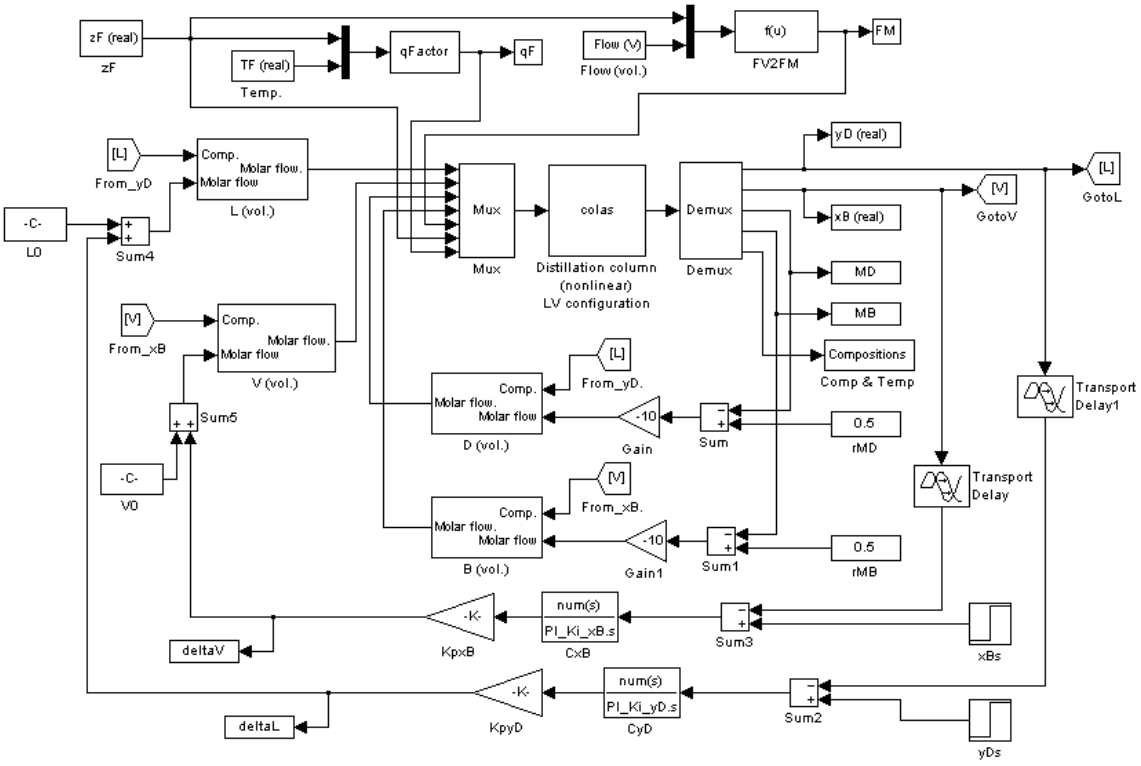


Figure 3.6.- Nonlinear distillation column Simulink model.

The Simulink model is fully parameterized to ease the creation of different scenarios. These parameters are stored in the "Model Workspace" object. A MATLAB interface has been implemented to ease the interaction with all these parameters and to store and manage the simulation results (see section 7.1). The configuration parameters are the following:

3.2.2.1. Sample time

The Simulink model is solved using MATLAB ordinary differential equations (ODE) solver with a variable-step method. The simulation results are stored into "To workspace" Simulink blocks (Figure 3.7). This block uses *Sample_time* parameter to indicate the elapsed time between measurements, independently of the step used for ODE solving. The time used for the

simulation is in minutes, so if we use a *Sample_time* = 0.5, the workspace variable associated to the Simulink block (that is, *FM* in this case) will store the results every 0.5 minutes.

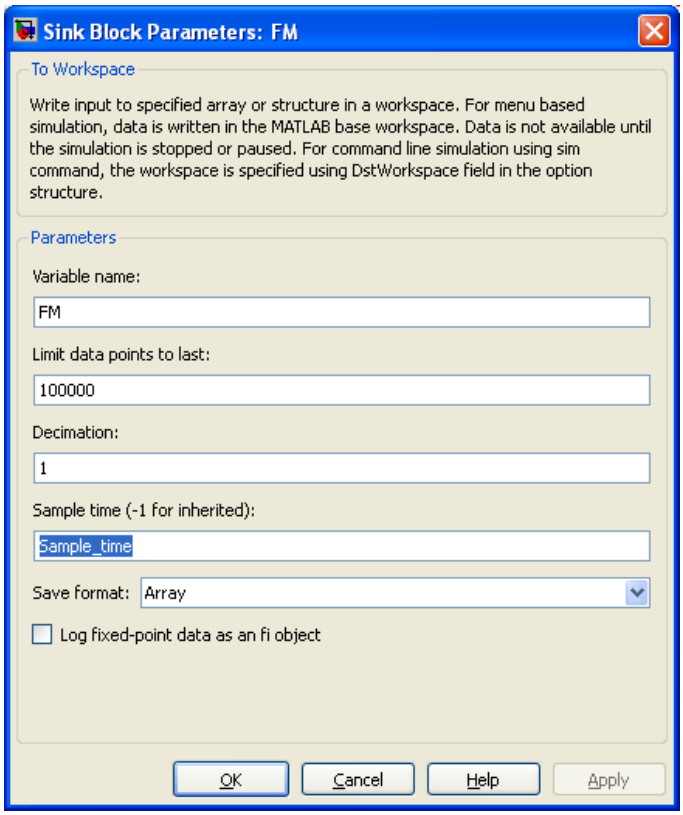


Figure 3.7.- "To workspace" block

3.2.2.2. Feed changes

If we have a look to Figure 3.8, we can see the subsystem block used to generate values for the feed temperature of the column. There are similar subsystems for the other measured disturbances: feed composition and feed flow rate.

Under normal operating conditions the parameter *ActivateDisturbances_T* = 0 and the signal used for the simulation is *SignalT*. When *ActivateDisturbances_T* = 1, the signal used for the simulation is *SignalDisturbT* (see section 7.2 for more details about the generation of the disturbance signals). This is used to simulate the column under abnormal conditions. Both signals are generated by code.

Under normal conditions, process equipments are subjected to little variations in their input variables. This situation is simulated through the use of several parameters in the Simulink model:

1. *ChangeRatio_TF*: change percentage in feed temperature. There are another two similar parameters for the change percentage in feed composition and in feed flow rate.
2. *ChangeRatio_Period*: indicates the sample time of the changes (this is the same for all three input variables: temperature, composition and flow rate).

The signal is generated from the nominal initial point (for example, 55ºC for feed temperature). Then, every *ChangeRatio_Period* minutes a random change ratio is applied. This is a random number between -*ChangeRatio_TF* and +*ChangeRatio_TF*. A delimiting zone is determined in order to keep process under normal operating conditions. In the example in Figure 3.9 this zone is twice the value of *ChangeRatio_TF*. If the new value is outside this zone it is forced to the limit value.

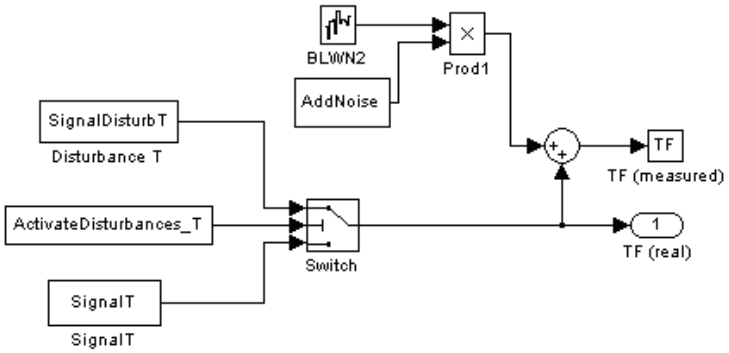


Figure 3.8.- "Feed temperature" subsystem block

The transition to one point to the other is done by a ramp instead of a pulse. This smoothes the change between two consecutive points and lets the simulation to fit gradually to changes.

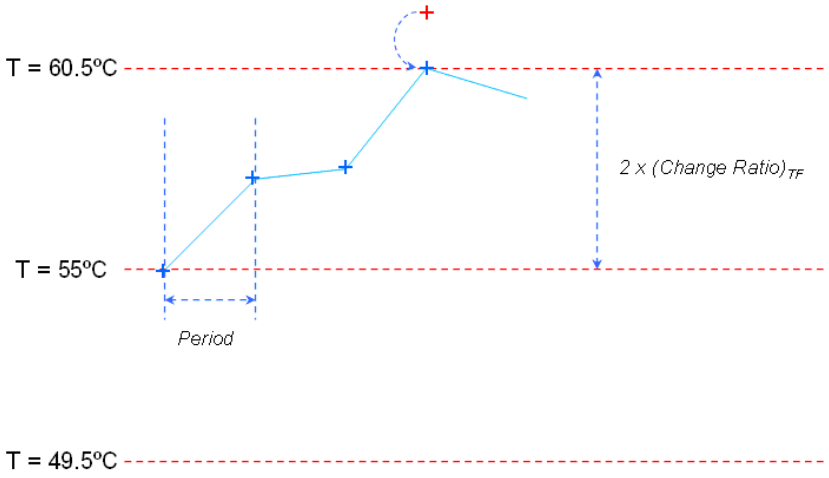


Figure 3.9.- Signal generation for feed temperature.

3.2.2.3. Noise

In order to obtain more realistic results, the model incorporates the possibility to add noise to the simulation measurements, as if they were captured through a plant information system. This is done via a "Band-Limited White Noise" source block (Figure 3.10). "Noise power" and "Sample time" parameters are explained in section 3.2.5

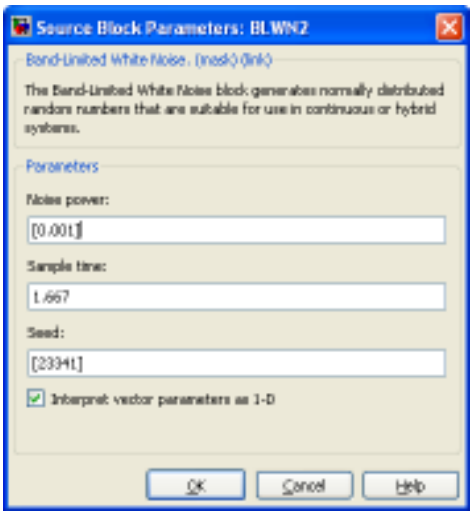


Figure 3.10.- "Band-Limited White Noise" source block.

3.2.2.4. Disturbances

Several types of disturbances have been implemented in the simulation model (Figure 3.11). We can specify the time at which the disturbance will be activated and the size, type (spike, ramp and pulse) and duration of the disturbance.

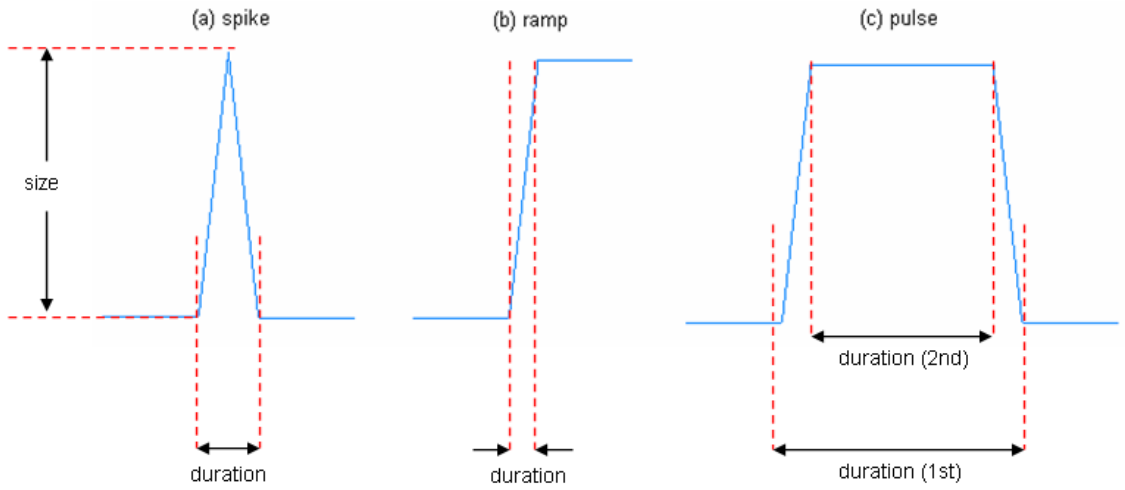


Figure 3.11.- Types of disturbances signals.

3.2.2.5. Regulatory control failures

As we will see in section 5.3 there are several types of failures related to regulatory controls. In this work we have simulated a failure control in both PI composition control loops x_B and y_D (see section 4.4.1). We simulate a failure in the control valve that modifies flows V (or L) to keep the set point of x_B (or y_D). The failure consists on a malfunction in the valve so it cannot be opened

till the position that the control proposes. This is done in Simulink via a saturation block that limits the flow (Figure 3.12). The saturation value is the mean value under normal operating conditions.

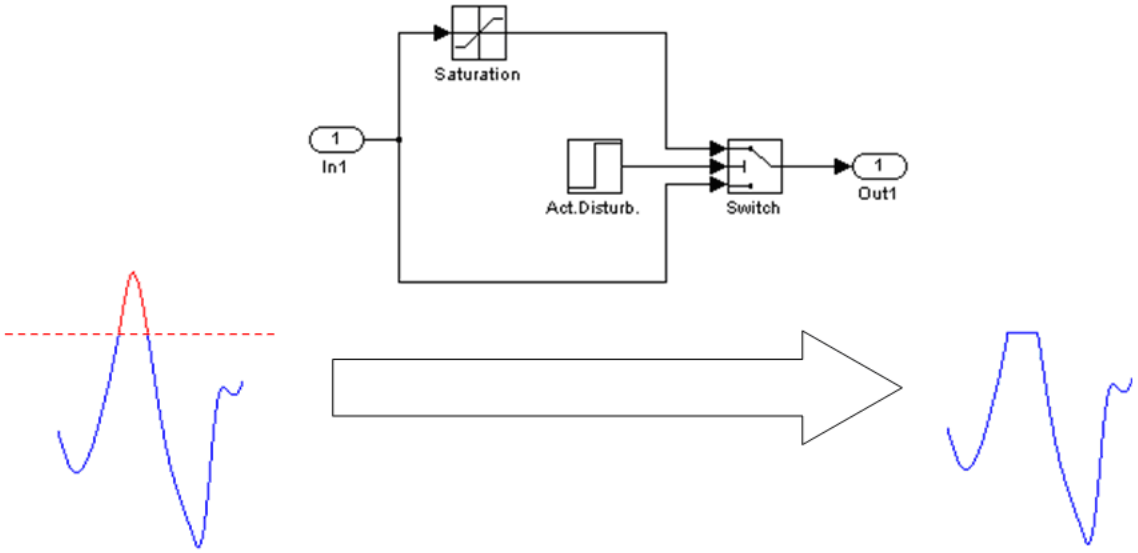


Figure 3.12.- Saturation block scheme for PI failure simulation.

3.2.2.6. Operating changes

The S-Function that simulates the column uses a mat-file to store some variables needed to initialize the simulation. This is like a snapshot of the column states at a given time, which allows starting a new simulation from that point.

From the original set point at x_B = 0.01 and y_D = 0.99, it has been explored another operating points that keep model stability. Step changes are included as the set points of x_B and y_D. No other changes or disturbances are done to the column and the simulation time is long enough to let it reach a steady state at the new operating point. Then, the steady values of the input variables, compositions and holdups are stored in the initialization mat-file. Now, the model is able to start a simulation from this new operating point, loading this new initialization file. This functionality is implemented with several model parameters (see section 7.1).

3.2.3. Calculation of *q* factor

The Simulink model implemented by Skogestad [1] uses the *q* factor as an input. In order to simulate a more realistic plant model, *q* factor will be calculated from composition and temperature measurements as shown in 2.2.4. This has been implemented in Simulink with an S-Function that has two inputs: feed composition (*z*) and feed temperature (*T*), and returns the corresponding *q* factor.

3.2.4. Molar and volumetric flows

The original model uses molar flow as an input. Again, the objective of this work is to simulate a plant model, where flow rates are typically measured as volumetric flow rates.

The feed is a saturated liquid mixture of methanol and ethanol with z_F composition. Supposing constant density, the relationship between molar and volumetric flows is the following:

$$F_M = \frac{F_V}{60}\left(z_F \frac{\rho_m}{M_m} + (1 - z_F)\frac{\rho_e}{M_e} \right) \qquad \textbf{Eq. 3.13}$$

where,

F_M: molar flow (kmol/min)

F_V: volumetric flow (L/h)

ρ_m: density of methanol (791.8 g/L, at 20ºC)

M_m: molar weight of methanol (32040 g/kmol)

ρ_e: density of ethanol (789.0 g/L, at 20ºC)

M_e: molar weight of ethanol (46070 g/kmol)

For example, if we want a molar flow of F_M = 1 kmol/min as an input for the distillation column when z_F = 0.5 then we need the following value of F_V :

$$F_V = \frac{1}{0.5 \times \frac{791.8}{32040} + 0.5 \times \frac{789.0}{46070}} \times 60 = 2868.14\ L/h$$

The flows *L*, *D* and *B* are in liquid state, so we can apply Eq. 3.13, but *V* is a vapor phase. We will assume an ideal gas, so the relationship between volume (*V*) and mol quantity (*n*) is:

$$\frac{V}{n} = \frac{R\,T}{P} \qquad \textbf{Eq. 3.14}$$

And volumetric flow can be calculated as:

$$F_V = \frac{F_M\ R\,T}{P} \times 6 \cdot 10^4 \qquad \textbf{Eq. 3.15}$$

where,

F_M: molar flow (kmol/min)

F_V: volumetric flow (L/h)

P: pressure (atm)

T: temperature (K)

R: gas constant, 0.082057 atm L / K mol

Temperature is obtained through interpolation in the *Txy* diagram.

The minimum dynamic pressure drop for a tray is about 25 mm H_2O/tray. The column has 41 theoretical trays, including reboiler and condenser, so it has 39 physical trays. Then the pressure drop in the column is:

$$P = 39\ \text{trays} \times 25\ \frac{\text{mm H}_2\text{O}}{\text{tray}} \times \frac{1\ \text{atm}}{10332.3\ \text{mm H}_2\text{O}} = 0.094364\ \text{atm}$$

Which is negligible compared to the column pressure.

The conversion for liquid phase is implemented in Simulink with simple block functions. The conversion for vapor phase implies an interpolation and an S-Function is needed to accomplish this.

3.2.5. White noise

As stated above, the model incorporates white noise through a "Band-Limited White Noise" source block (Figure 3.10). This Simulink block has two parameters. The first one is the "Noise power", that is, the value of the white noise, which is around 10^{-5} times smaller than the value of the signal to be modified. For example, the nominal value for the feed temperature is 55ºC, and then its associated white noise will be around $55 \times 10^{-5} \approx 10^{-3}$

On the other hand, the correlation time of the noise generated by this Simulink block is the "Sample rate" parameter of the block. For accurate simulations, it is advisable to use a correlation time (t_c) much smaller than the fastest dynamics of the system. The following equation yields good results:

$$t_c = \frac{2\pi}{100\ f_{\max}} \qquad \textbf{Eq. 3.16}$$

where f_{max} is the bandwidth of the system in rad/s

The dynamics of most distillation columns are dominated by one large time constant (τ_C), which is nearly the same, regardless of where a disturbance is introduced or where composition is measured. Physically, this dominant time constant reflects the change in component holdup inside the column. The dominant time constant estimated for the column is around 25 min [(7)].

Then, $f_{\max} = \frac{1}{60 \times 25} = 0.0006283\ rad/s$ and $t_c = \frac{2\pi}{100 \times 0.0006283} \approx 100\ s = 1.667\ \text{min}$

3.3. Process exploratory analysis

3.3.1. Introduction

Design of Experiments (DOE) is used to get a better process understanding. The objective is to probe how the column "reacts" to input disturbances and to check model stability and possible non-linearities. In chapter 0 major troubleshooting in distillation columns is considered, but, by now, we will only take into account disturbances in feed characteristics, *i*.e., input variables z_F (feed composition), T_F (feed temperature) and F (feed flow rate). The study is run at two levels, so we have a 2^3 factorial design.

3.3.2. Feed changes

The factors of the study are the input variables to the system, that is:

Table 3.1.- DOE factors

Factor	Units	Value
F	L/h	2868.14
T_F	ºC	55
z_F	Molar composition	0.5

The objective is to explore the zone around the steady state conditions for the operating point described in section 2.2.5. Under normal operating conditions, we have assumed feed changes around a 5% ratio, so this is the space that we will explore with DOE: +/- 5% around the operating point. The values for the experimental runs are shown in Table 3.2.

Table 3.2.- Values of factors levels.

Factor	Low (-5%)	High (+5%)
F	2724.73	3011.55
T_F	52.25	57.75
z_F	0.475	0.525

The response variables will be L, V, D and B, which are the flows manipulated to control the variables x_B, M_B, y_D and M_D.

An m-file is created in MATLAB to run all the combinations of the factorial plan. In every run, there is enough time for the simulation to reach the steady state. Table 3.3 shows the values of the observed variables obtained in every simulation.

Table 3.3.- Results of the 2^3 factorial design.

Plan				Results			
#	*F*	*T*	*z*	*L* (m^3/h)	*V* (10^3 m^3/h)	*D* (m^3/h)	*B* (m^3/h)
1	-1	-1	-1	6.16066	5.22467	1.08788	1.72560
2	-1	-1	1	6.26338	5.39383	1.22685	1.58660
3	-1	1	-1	6.17528	5.21179	1.08788	1.72562
4	-1	1	1	6.27684	5.37999	1.22683	1.58663
5	1	-1	-1	6.80823	5.77392	1.20232	1.90713
6	1	-1	1	6.92175	5.96090	1.35595	1.75346
7	1	1	-1	6.82438	5.75967	1.20231	1.90714
8	1	1	1	6.93665	5.94561	1.35592	1.75349

These data are analyzed with Statgraphics Centurion XV. The first ANOVA analysis includes all the effects to see their relative importance, which could be done with two useful tools: the normal probability plot of the effects and the pareto chart of the effects.

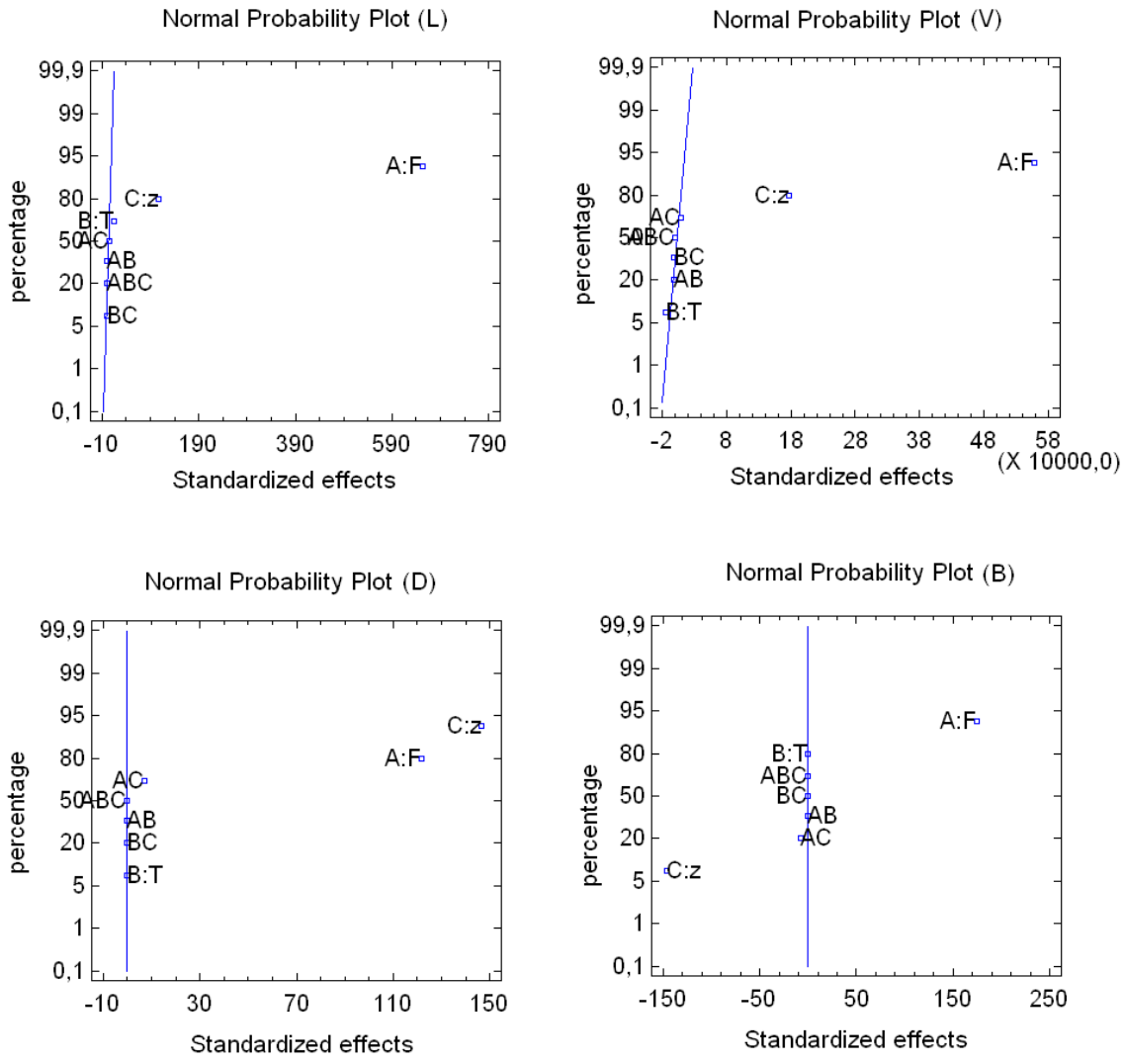


Figure 3.13.- Normal probability plot of the effects (first analysis).

Normal probability plot (npp) of the effects for the four responses analyzed are shown in Figure 3.13. Null effects will be estimated from normal random effects with zero mean, and thus they will approximately be lying on the npp through a straight line crossing the point (0.50%). On the other hand, actual effects will be estimated from normal random effects with positive (or negative) mean, and thus they will appear on the right upper (or left lower) part of the normal probability plot. This is a very helpful quick first screen or assessment of what is going on in the data.

If we have a look at Figure 3.13 we see that simple effects A and C (flow rate and composition of the feed, respectively) are clearly significant for every studied response variable, and, in the case of response variables D and B, the interaction effect AC is slightly significant.

Another way to see the relative importance of the effects is the pareto chart (Figure 3.14) which contains a bar for each standardized effect, sorted from absolute value and coloring from signs. The length of each bar is proportional to the standardized effect, which equals the magnitude of the t-statistic that would be used to test the statistical significance of that effect. As this analysis contains all effects, no degrees of freedom are available to estimate errors, and the pareto chart in Figure 3.14 plots the value of the effects (not standardized effect). Those errors will be computed in a second analysis, in which the lower effects detected in the previous analysis are not considered anymore. From Figure 3.14 the same significant effects detected in Figure 3.13 are obtained.

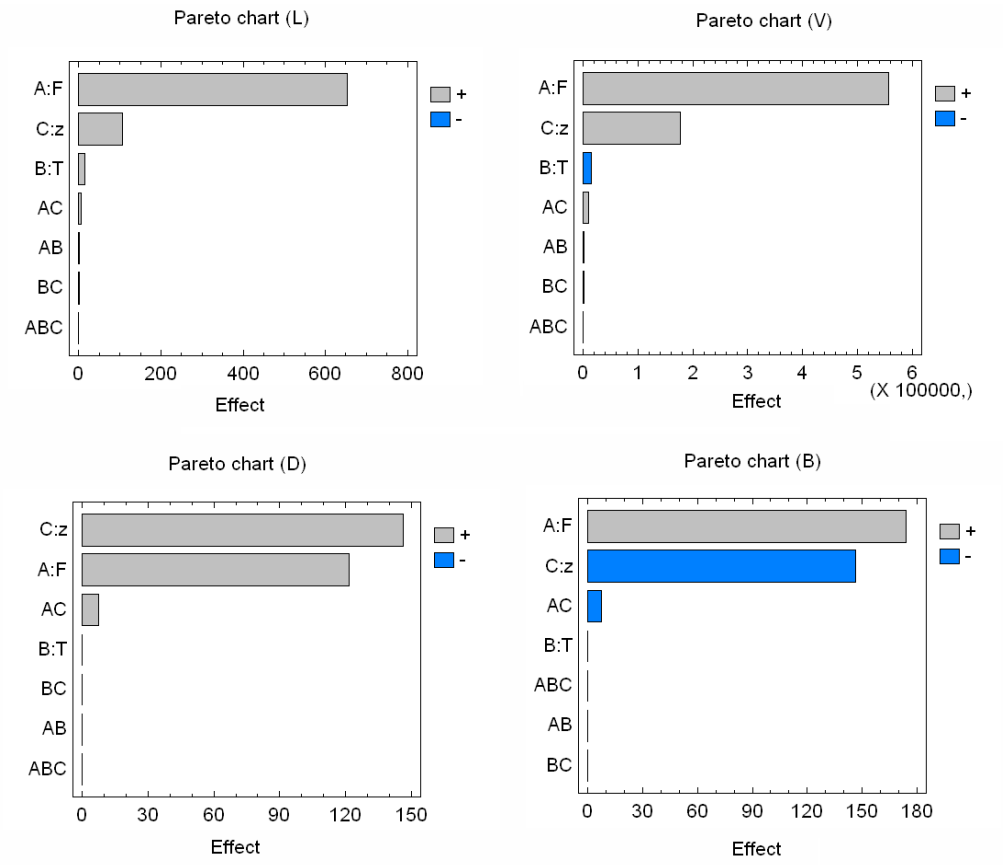


Figure 3.14.- Pareto charts

Table 3.4 shows the effects taken into account for the second analysis. If we plot the pareto chart for this second analysis (Figure 3.15) we see a vertical line that indicates the location of the 0.05 critical value for Student's *t*. Any bars that extend to the right of that line indicate effects that are statistically significant at the 5% significance level.

Table 3.4.- Effects considered for the second analysis

Variable	Effect
L	A, C
V	A, C
D	C, A, AC
B	A, C, AC

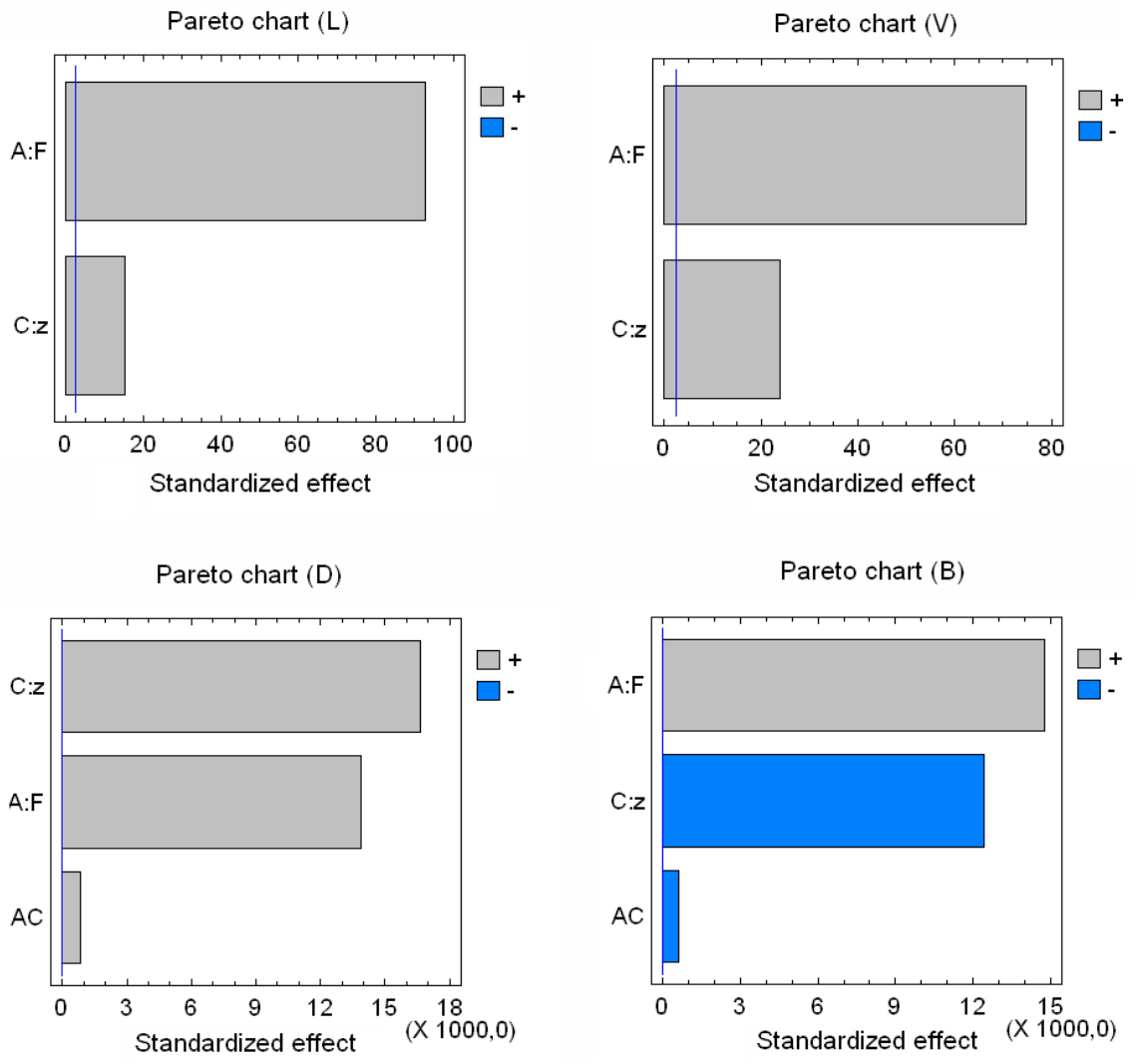


Figure 3.15.- Pareto charts for the second analysis

Exact P-values may also be obtained from the ANOVA table (Table 3.5 to Table 3.8):

Table 3.5.- Analysis of Variance for L

Source	*Sum of Squares*	*Df*	*Mean Square*	*F-Ratio*	*P-Value*
A:F	854669,	1	854669,	8607,65	0,0000
C:z	23119,1	1	23119,1	232,84	0,0000
Total error	496,459	5	99,2919		
Total (corr.)	878285,	7			

Table 3.6.- Analysis of Variance for V

Source	*Sum of Squares*	*Df*	*Mean Square*	*F-Ratio*	*P-Value*
A:F	6,21513E11	1	6,21513E11	5594,34	0,0000
C:z	6,30623E10	1	6,30623E10	567,63	0,0000
Total error	5,55484E8	5	1,11097E8		
Total (corr.)	6,85131E11	7			

Table 3.7.- Analysis of Variance for D

Source	*Sum of Squares*	*Df*	*Mean Square*	*F-Ratio*	*P-Value*
A:F	29652,6	1	29652,6	19206184	0,0000
C:z	42803,5	1	42803,5	27724134	0,0000
AC	107,522	1	107,522	696429,10	0,0000
Total error	0,000617563	4	0,000154391		
Total (corr.)	72563,6	7			

Table 3.8.- Analysis of Variance for B

Source	*Sum of Squares*	*Df*	*Mean Square*	*F-Ratio*	*P-Value*
A:F	60686,2	1	60686,2	21854665	0,0000
C:z	42823,9	1	42823,9	15421964	0,0000
AC	107,527	1	107,527	387231,46	0,0000
Total error	0,00111072	4	0,000277681		
Total (corr.)	103618,	7			

If we have a look at Figure 3.16, we can see the effects of feed flow rate (*F*) and composition (*z*) in each studied response variable.

Obviously, if feed flow rate increases, all flows (*L*, *V*, *D* and *B*) increase.

The effect of feed composition is not so obvious. The distillates composition y_D is controlled by manipulating flow *L* and the bottoms composition x_B is controlled with flow *V*. To keep set points, when feed composition increases the PI control systems compensates this by increasing both flows V and L. On the other hand, an increase in feed composition means that the feed enriches in the more volatile compound, so there is more product in the distillate and the level in the distillate drum tends to increase. To keep control level, the proportional control increases the distillate flow *D* (the effect of composition feed *z* in *D* is negative, see Figure 3.16)

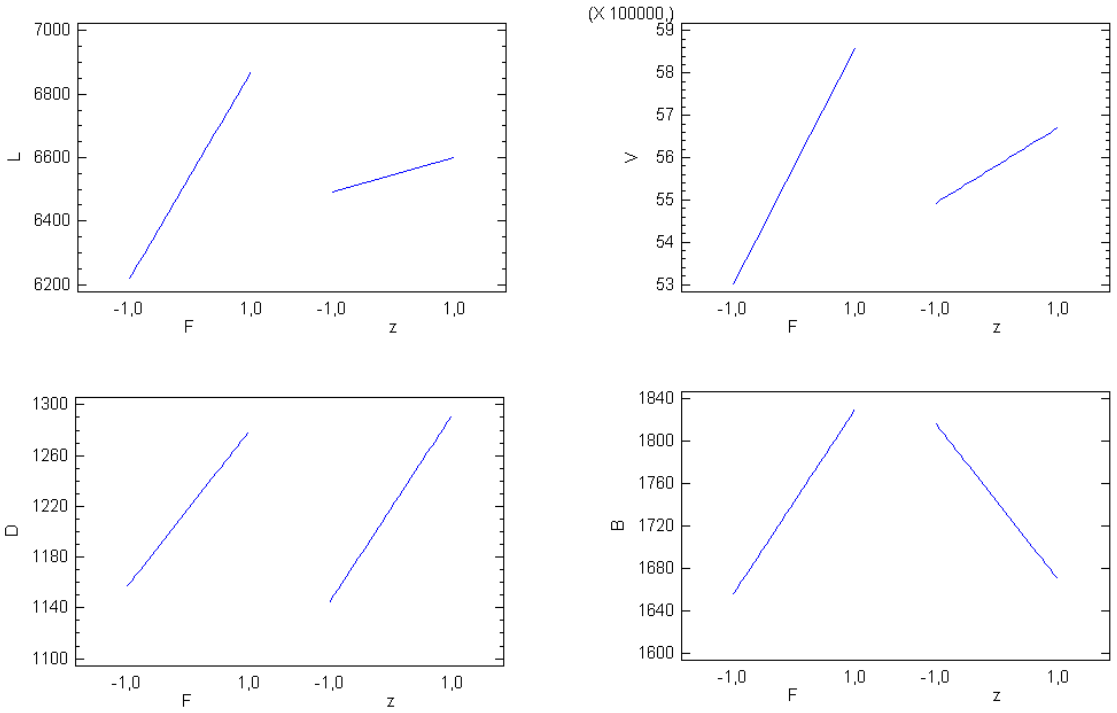


Figure 3.16.- Main effects plot.

If more volatile compound enters to the column, then the flow richer in the more volatile compound (e.g. distillate *D*) will increase and the flow richer in heavy compound (e.g. bottoms *B*) will decrease.

Figure 3.17 shows the interaction plot for D and B variables. We can see that those interactions are very small (lines are almost parallel).

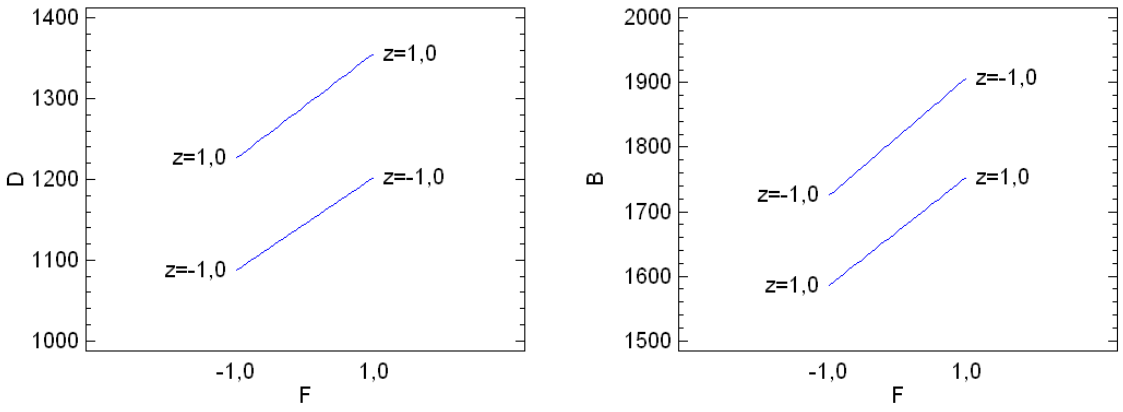


Figure 3.17.- Interaction plots for D and B.

4. Process dynamics and control

4.1. Introduction

A process is the conversion of feed materials to products using chemical and physical operations. In practice, the term process tends to be used for both the processing operation and the processing equipment.

The primary objective of process control is to maintain a process at the desired operating conditions, safely and efficiently, while satisfying environmental and product quality requirements. The subject of process control is concerned with how to achieve these goals. In large-scale, integrated processing plants such as oil refineries or ethylene plants, thousands of process variables such as compositions, temperatures, and pressures are measured and must be controlled. Fortunately, large numbers of process variables (mainly flow rates) can usually be manipulated for this purpose. Feedback control systems compare measurements with their desired values and then adjust the manipulated variables accordingly [(10)].

The process control problem is characterized by identifying three important types of process variables:

- Controlled variables (*CVs*). The process variables that are controlled. The desired value of a controlled variable is referred to as set point.
- Manipulated variables (*MVs*): The process variables that can be adjusted to keep the controlled variables at or near their set points. Typically, the manipulated variables are flow rates.
- Disturbance variables (*DVs*): Process variables that affect the controlled variables but cannot be manipulated. Disturbances generally are related to changes in the operating environment of the process, for example, its feed conditions or ambient temperature. Some disturbance variables can be measured on-line, but many others cannot.

The specification of *CVs*, *MVs*, and *DVs* is a critical step in developing a control system. The selections should be based on process knowledge, experience and control objectives.

4.2. Feedback and feed forward control

A system that positions the manipulated variable either manually or on a programmed basis without using any process measurements is called an open-loop system. This operation is acceptable for well-defined processes without disturbances.

On the other hand, a closed-loop system uses the measurement of one or more process variables to move the manipulated variable to achieve control. Closed-loop systems may include feed forward, feedback or inferential control strategies, or any combination of them [11].

A feedback controller uses direct measurements of the controlled variables to adjust the values of the manipulated variables. The objective is to keep the controlled variables at desired levels (set points).

For example, consider a simple distillation column separating a binary mixture of pentane and hexane into two product streams of pentane (distillate) and hexane (bottoms). The control objective is to maintain the production of a distillate stream with 95 mole % pentane in the presence of changes in the feed composition. We could use a composition analyzer to measure the concentration of pentane in the distillate and then, using feedback control, it is possible to manipulate the reflux ratio, so that we can keep the distillate 95% in pentane (Figure 4.1).

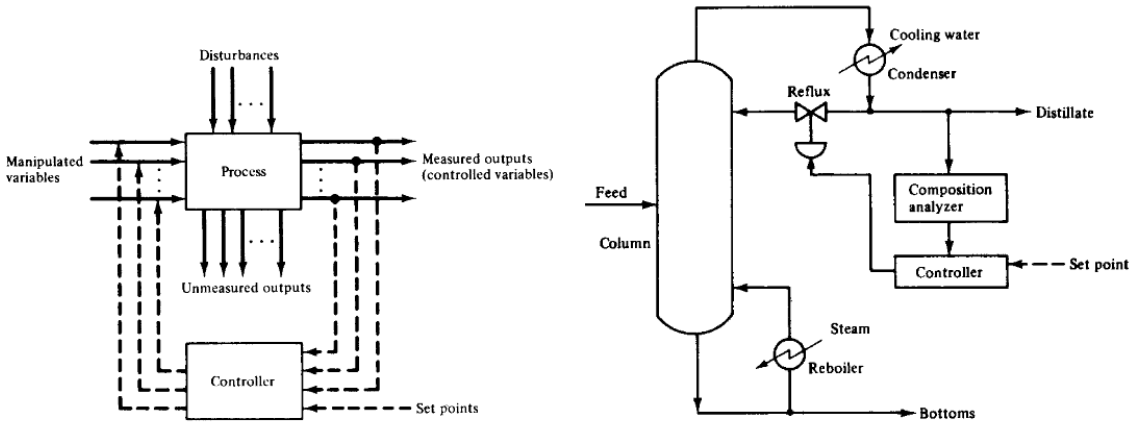


Figure 4.1.- Feedback control configuration [11]

An important advantage of feedback control is that corrective action occurs regardless of the source of the disturbance. Its ability to handle disturbances of unknown origin is a major reason why feedback control is the dominant process control strategy. Another important advantage is that feedback control reduces the sensitivity of the controlled variable to unmeasured disturbances and process changes. However, feedback control does have a fundamental limitation: no corrective action is taken until after the disturbance has upset the process, that is, until after the controlled variable deviates from the set point.

An alternative control system is a feed forward control configuration, which uses direct measurement of the disturbances to adjust the values of the manipulated variables. The objective here is to keep the values of the controlled output variables at desired levels. In the distillation example, we can use a composition analyzer to monitor the concentration of pentane in the feed. Then in a feed forward arrangement we can change the reflux ratio to achieve our objective (Figure 4.2).

The important advantage of feed forward control is that corrective action is taken before the controlled variable deviates from the set point. Ideally, the corrective action will cancel the effects of the disturbance so that the controlled variable is not affected by the disturbance. Although ideal cancellation is generally not possible, feed forward control can significantly reduce the effects of measured disturbances.

Feed forward control has three significant disadvantages:

1. The disturbance variable must be measured (or accurately estimated).
2. No corrective action is taken for unmeasured disturbances.
3. A process model that relates measured disturbances and controlled variables is required.

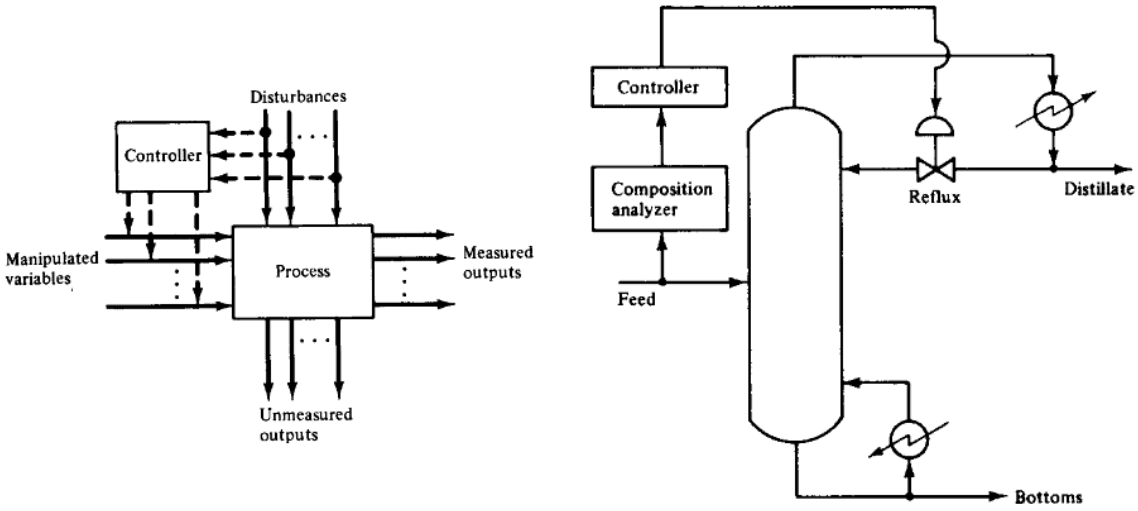


Figure 4.2.- Feed forward control configuration [11]

In industrial applications it is generally uneconomical to attempt to measure all potential disturbances. A more practical approach is to use a combined feed forward-feedback control system, where feedback control provides corrective action for unmeasured disturbances, while feed forward control reacts to eliminate measured disturbances before the controlled variable is upset. Consequently, in industrial applications, feed forward control is normally used in combination with feedback control.

There are three basic control modes: proportional, integral and derivative [12].

4.2.1. Proportional control

In feedback control, the objective is to reduce the error signal to zero where

$$e(t) = y_{sp}(t) - y_m(t) \qquad \textbf{Eq. 4.1}$$

and

$e(t)$ = error signal

$y_{sp}(t)$ = set point

$y_m(t)$ = measured value of the controlled variable

Although this equation indicates that the set point can be time-varying, in many process control problems it is kept constant for long periods of time.

For proportional control, the controller output is proportional to the error signal,

$$p(t) = \bar{p} + K_c e(t) \qquad \textbf{Eq. 4.2}$$

where

$p(t)$ = controller output

$\bar{p}$ = bias (steady-state) value

K_c = controller gain (usually dimensionless)

The key concepts behind proportional control are the following:

- The controller gain can be adjusted to make the controller output changes as sensitive as desired to deviations between se point and controlled variable
- The sign of K_c can be chosen to make the controller output increase (or decrease) as the error signal increases.

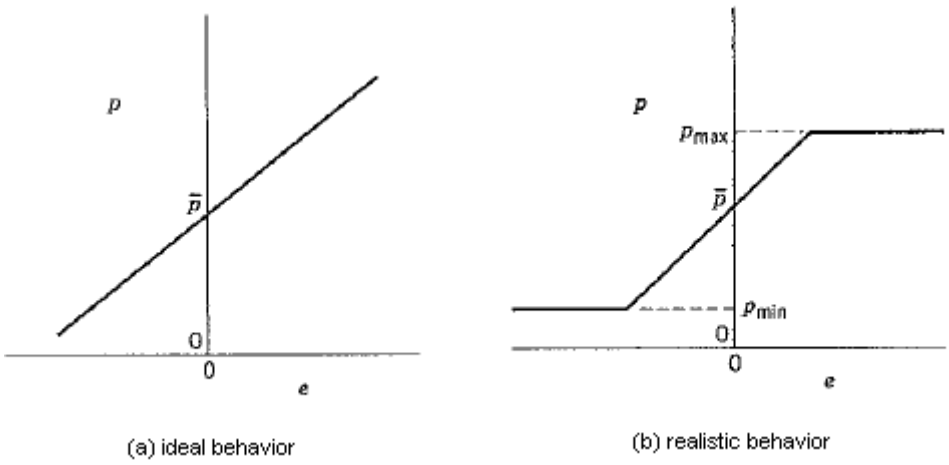


Figure 4.3.- Proportional control: ideal vs. realistic behavior [12]

The ideal proportional controller in Figure 4.3a does not include physical limits on the controller output. A more realistic representation is shown in Figure 4.3b, where the controller saturates when its output reaches a physical limit, either p_{max} or p_{min}. In order to derive the transfer

function for an ideal proportional controller (without saturation limits), define a deviation variable *p'(t)* as

$$p'(t) = p(t) - \bar{p} \qquad \textbf{Eq. 4.3}$$

Then Eq. 4.2 can be written as

$$p'(t) = K_c e(t) \qquad \textbf{Eq. 4.4}$$

An inherent disadvantage of proportional-only control is that a steady-state error (or *offset*) occurs after a set-point change or a sustained disturbance. In principle, offset can be eliminated by manually resetting the set point y_{sp}. However, this approach is inconvenient because operator intervention is required and the new value of y_{sp} must usually be found by trial and error. In practice, it is more convenient to use a controller that contains integral control action. The integral control mode provides automatic reset, as discussed below.

For control applications where offsets can be tolerated, proportional-only control is attractive because of its simplicity. For example, in some level control problems like in the reboiler and condenser, maintaining the liquid level close to the set point is not as important as merely ensuring that the storage tank does not overflow or run dry.

4.2.2. Integral control

For integral control action, the controller output depends on the integral of the error signal over time,

$$p(t) = \bar{p} + \frac{1}{\tau_I}\int_0^t e(t^*)\, dt^* \qquad \textbf{Eq. 4.5}$$

, where τ_I is an adjustable parameter referred to as the integral time or reset time and has units of time.

Integral control action is widely used because it provides an important practical advantage, the elimination of offset. To understand why offset is eliminated, consider Eq. 4.5. In order for the process being controlled to be at steady state, the controller output *p* must be constant so that the manipulated variable is also constant. Eq. 4.5 implies that *p* changes with time unless *e(t*)*=0. Thus when integral action is used, *p* automatically changes until it attains the value required to make the steady-state error zero. This desirable situation always occurs unless the controller output or final control element saturates and thus is unable to bring the controlled variable back to the set point. Controller saturation occurs when the disturbance or set-point change is so large that it is beyond the range of the manipulated variable.

Although elimination of offset is usually an important control objective, the integral controller in Eq. 4.5 is seldom used by itself because little control action takes place until the error signal has

persisted for some time. In contrast, proportional control action takes immediate corrective action as soon as an error is detected. Consequently, integral control action is normally used in conjunction with proportional control as the proportional-integral (PI) controller:

$$p(t) = \bar{p} + K_c\left(e(t) + \frac{1}{\tau_I}\int_0^t e(t^*)\,dt^*\right) \qquad \textbf{Eq. 4.6}$$

The response of the PI controller to a unit step change in *e(t)* is shown in Figure 4.4. At time zero, the controller output changes instantaneously due to the proportional action. Integral action causes the ramp increase in *p(t)* for $t > 0$. When $t = \tau_I$, the integral term has contributed the same amount to the controller output as the proportional term. Thus, the integral action has repeated the proportional action once. Some commercial controllers are calibrated in terms of $1/\tau_I$ (repeats per minute) rather than τ_I (minutes, or minutes per repeat). For example, if τ_I =0.2 min, this corresponds to $1/\tau_I$ having a value of 5 repeats/minute.

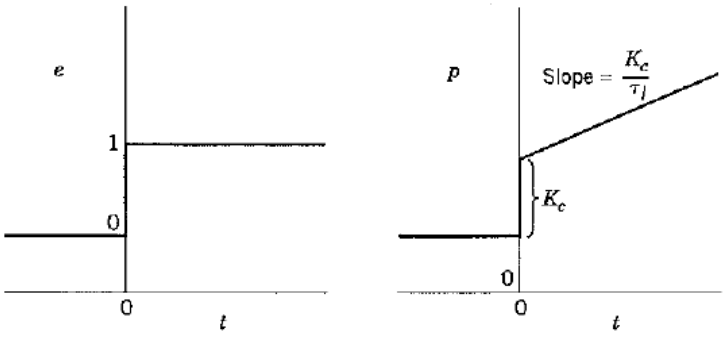


Figure 4.4.- Response of proportional-integral controller to unit step change (12)

One disadvantage of using integral action is that it tends to produce oscillatory responses of the controlled variable and reduces the stability of the feedback control system. A limited amount of oscillation can usually be tolerated because it is often associated with a faster response. The undesirable effects of too much integral action can be avoided by proper tuning of the controller or by including derivative action that tends to counteract the destabilizing effects

4.2.3. Derivative control

The function of derivative control action is to anticipate the future behavior of the error signal by considering its rate of change. For example, suppose that a reactor temperature increases by 10°C in a short period of time, say 3 min. This clearly is a more rapid increase in temperature than a 10°C rise in 30 min, and it could indicate a potential runaway situation for an exothermic reaction. If the reactor were under manual control, an experienced plant operator would anticipate the consequences and quickly take appropriate corrective action to reduce the temperature. Such a response would not be obtainable from the proportional and integral control modes discussed so far. Note that a proportional controller reacts to a deviation in

temperature only, making no distinction as to the time period over which the deviation develops. Integral control action is also ineffective for a sudden deviation in temperature because the corrective action depends on the duration of the deviation.

The anticipatory strategy used by the experienced operator can be incorporated in automatic controllers by making the controller output proportional to the rate of change of the error signal or the controlled variable. Thus, for ideal derivative action,

$$p(t) = \overline{p} + \tau_D \frac{d\,e(t)}{dt} \qquad \textbf{Eq. 4.7}$$

where τ_D, the derivative time, has units of time. Note that the controller output is equal to the nominal value $\overline{p}$ as long as the error is constant (that is, as long as *de(t)/dt* = 0). Consequently, derivative action is never used alone; it is always used in conjunction with proportional or proportional-integral control. By providing anticipatory control action, the derivative mode tends to stabilize the controlled process. Thus, it is often used to counteract the destabilizing tendency of the integral mode.

4.2.4. Proportional-Integral-Derivative Control

The combination of the proportional, integral, and derivative control modes as a PID controller could be done in many variations. The three most common forms are the parallel and series forms (Figure 4.5).

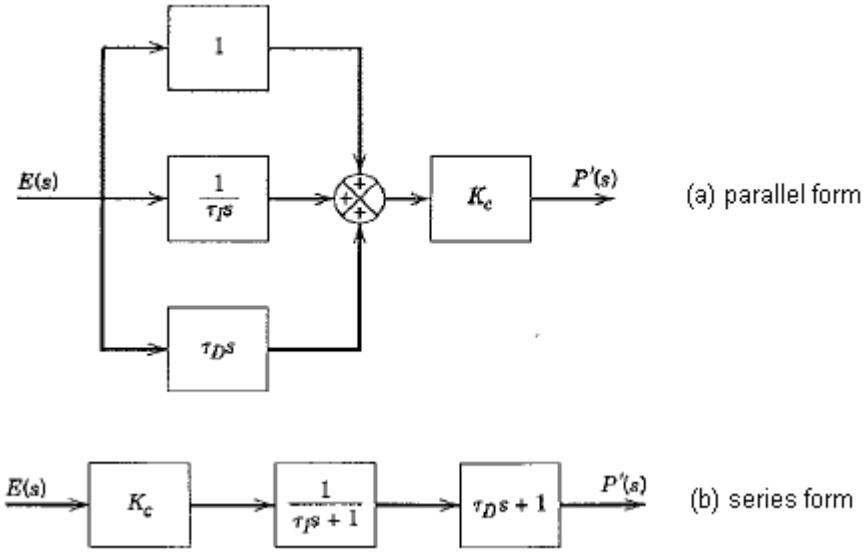


Figure 4.5.- Block diagram of the parallel and series forms of PID control [12]

PID controllers are used in loops where signals are not noisy and where tight dynamic response is important. The derivative action helps to compensate for lags in the loop. Temperature

controllers in reactors are usually PID. The controller senses the rate of movement away from the set-point and starts moving the control valve earlier than with only PI action.

4.2.5. Typical responses of feedback control systems

The responses shown in Figure 4.6 illustrate the typical behavior of a controlled process after a step change in a disturbance variable occurs. The controlled variable y represents the deviation from the initial steady-state value. If feedback control is not used, the process slowly reaches a new steady state. Proportional control speeds up the process response and reduces the offset. The addition of integral control action eliminates offset but tends to make the response more oscillatory. Adding derivative action reduces both the degree of oscillation and the response time. The use of P, PI, and PID controllers does not always result in oscillatory process responses; the nature of the response depends on the choice of the controller settings (K_c, τ_I and τ_D) and the process dynamics. However, the responses in Figure 4.6 are representative of what occurs in practice.

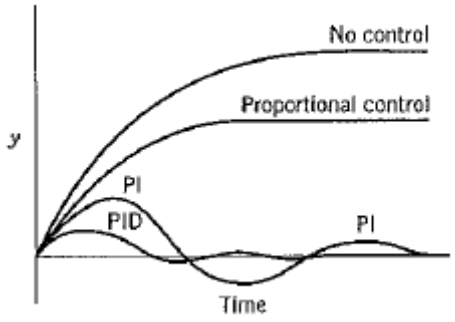


Figure 4.6.- Typical process responses with feedback control [12]

4.3. Inferential control

The function of composition control extends beyond assuring adequate product purity. The composition controller manipulates a stream such as reboil, reflux or product. Unstable composition control will disturb this stream, and the disturbances will unsettle the column. Two primary methods are used for composition control: temperature control or analyzer control [13].

Although gas chromatographs (GC's) are available to obtain on-line analysis of product samples, they are seldom used in distillation column applications because they are expensive to buy and to maintain and, most importantly, they provide delayed measurements, due both to sample time and dead time [14]. The delay introduced by a GC can be detrimental from a process control standpoint. Other measurement techniques such as infrared-based measurements, analysis of refractive index, density or dielectric constant are not yet reliable or accurate enough for use in distillation applications.

When quality indexes are too difficult or too expensive to measure in real time, sometimes it is possible to make use of secondary variables, easily measured in real time (e.g., pressures, temperatures, flow rates) and build a mathematical model that correlates these parameters and the variables that must be monitored. This technology is known as *soft sensors*. When controlled variables cannot be measured or measurements are too expensive, in some cases, there is the possibility to estimate them from other secondary variables. The inferential control configuration uses those secondary measurements to adjust the values of the manipulated variables (Figure 4.7). The objective here is to keep the (unmeasured) controlled variables at desired levels. The estimator (also known as *soft sensor*) uses the values of the available measured outputs, together with the material and energy balances that govern the process, to mathematically compute (estimate) the values of the unmeasured controlled variables. These estimates, in turn, are used by the controller to adjust the values of the manipulated variables. In our example, both control systems noted above (feedback and feed forward) depend on the composition analyzers. It is possible that such measuring devices are either very costly or of very low reliability for an industrial environment (failing quite often or not providing accurate measurements). In such cases we can measure the temperature of the liquid at various trays along the length of the column quite reliably, using simple thermocouples. Then using the material and energy balances around the trays of the column and the thermodynamic equilibrium relationships between liquid and vapor streams, we can develop a mathematical relationship that gives us the composition of the distillate if the temperatures of some selected trays are known. Figure 4.7 (right) shows such a control scheme that uses temperature measurements (secondary measurements) to estimate or infer the composition of pentane in the distillate (*i*.e., the value of the control objective). When it is not possible to obtain a mathematical model of the system from first principles, we can use different statistical approaches -like Partial Least Squares method [15] - to estimate the values of the unmeasured controlled variables.

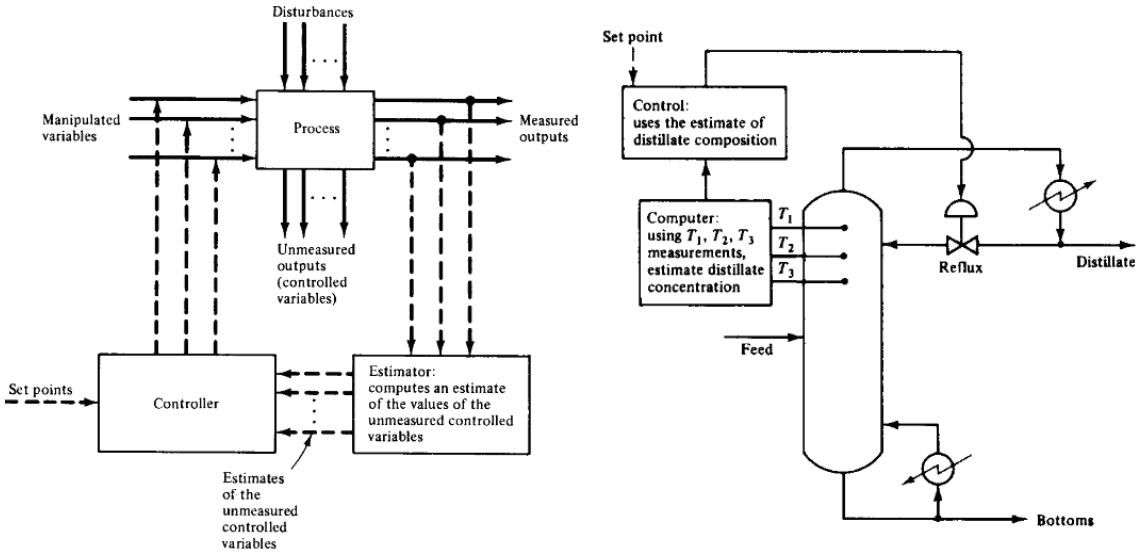


Figure 4.7.- Inferential control configuration [13]

In the case of distillation columns, column temperature control is perhaps the most popular means of controlling product composition. The control temperature is used as a substitute to product composition analysis. A change in control temperature represents a corresponding variation in the concentration of key components in the product. For instance, a rise in top section control temperature represents a rise in the concentration of the heavy key component in the top product. Temperature control is an easy and inexpensive means of controlling product composition. It uses a high-reliability, low-maintenance measuring element that suffers from little dynamic lag and downtime. These advantages make temperature control far more popular than analyzer control.

4.4. Multiloop control

4.4.1. Distillation column control

Consider the distillation column in Figure 4.8 that has five controlled variables and five manipulated variables. The controlled variables are product compositions, y_D and x_B, column pressure, p, and the liquid holdups in the reflux drum and reboiler, M_D and M_B, respectively [6]:

$$y = (x_D\ x_B\ M_D\ M_B\ p)^T$$

The five manipulated variables are product flow rates (D and B) and internal flow rates in the top (L, V_T) and the bottom of the column (V):

$$u = (L\ V\ D\ B\ V_T)^T$$

The feed stream is assumed to come from an upstream unit. Thus, the feed flow rate cannot be manipulated, but it can be measured and used for feed forward control. Other disturbances are temperature and composition of the feed.

A conventional multiloop control strategy for this distillation column would consist of five feedback control loops. Each control loop uses a single manipulated variable to control a single controlled variable. But, how should the controlled and manipulated variables be paired? The total number of different multiloop control configurations that could be considered is 5! = 120. Many of these control configurations are impractical or unfeasible such as any configuration that attempts to control reboiler holdup M_B by manipulating distillate flow D, for example. However, even after the unfeasible control configurations are eliminated, there are still many reasonable configurations left [15].

In almost all industrial control configurations, the distillation column is first stabilized by closing three decentralized (SISO) loops for level and pressure, involving the outputs

$$y_2 = (M_D\ M_B\ p)^T$$

The three SISO loops for controlling y_2 are the following:

1. Distillate holdup level (M_D) controlled by distillate flow (D).
2. Bottoms holdup level (M_B) controlled by bottoms flow (B).
3. Column pressure (p) controlled by condenser vapor flow (V_T).

The remaining outputs are then the product compositions

$$y_1 = (y_D \; x_B)^T$$

These SISO loops usually interact weakly and may be tuned independently of each other, that is, they are decentralized. However, since each vessel (reboiler and top drum separator) has an inlet and two outlet flows, there exists many possible choices for u_2 (and thus for u_1). By convention, each of these possible choices ("configuration") is named by the inputs u_1 left for composition control. For example, the LV-configuration (Figure 4.8) refers to a partially controlled system with

$$u_1 = (L \; V)^T; \quad u_2 = (D \; B \; V_T)^T$$

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

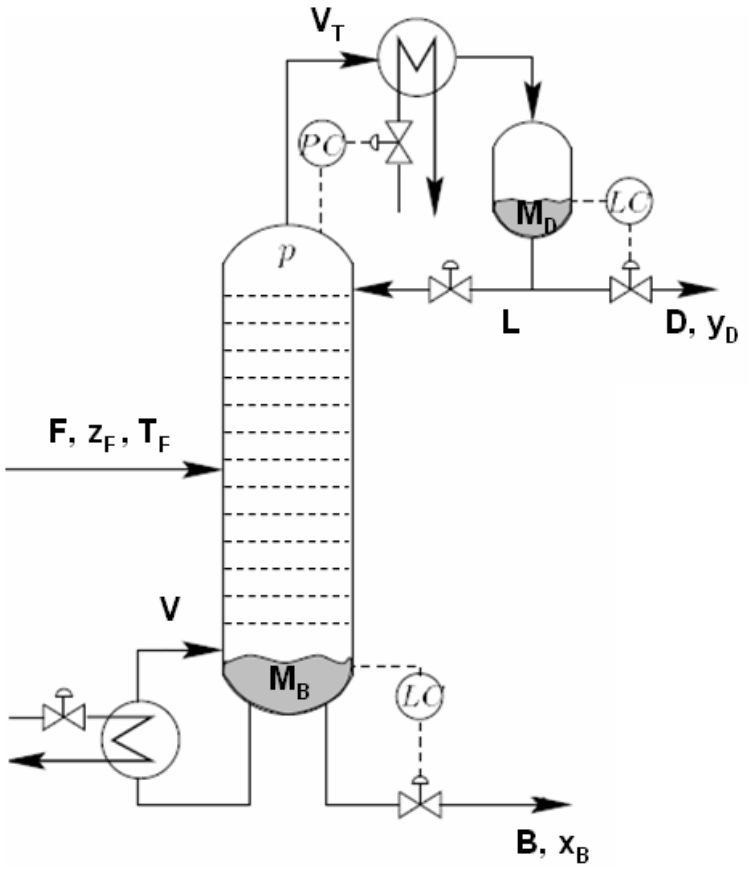


Figure 4.8.- Controlled and manipulated variables for a typical distillation column.

Another common configuration is the DV-configuration with

$$u_1 = (D\ V)^T; \quad u_2 = (L\ B\ V_T)^T$$

It is clear that we need to control the levels and pressure (y_2) to stabilize the system. It also seems reasonable that we should control both the product compositions (y_1), or at least some related quality variables, because, after all, the reason for having the distillation column in the first place is to split the feed stream into two products with different composition. However, in practice, this is often not the case. Therefore, we need for composition control to consider three different cases:

1. Open-loop: no composition control; the operators manually adjust the two remaining flows (u_1). The term "open-loop", which here refers to the composition control problem, is not quite correct since we assume there is already a level and pressure control system in place involving u_2 and y_2.
2. One-point control: one composition loop is closed.
3. Two-point control: both composition loops are closed

The case with no control ("open-loop") has been common in industrial practice. This is somewhat surprising, since without composition control, the distillation almost behaves as a large mixing tank, and if we leave the compositions uncontrolled, the columns will tend to drift away and be "filled up" of light or heavy component. In conclusion, from a practical point of view, the composition profile behaves almost like an "unstable" system. Thus, the composition profile needs to be continuously monitored to maintain stable operation, and this puts a heavy burden on the operators.

To "stabilize" the composition profile, it is therefore recommended to close at least one composition loop (one-point control, e.g. involving V and x_B). In this case the remaining degree of freedom (e.g. L) is adjusted manually.

From an economic point of view, disregarding the control and measurement problems, two-point control is the best. This follows since the optimal operating point generally corresponds to a given purity specification. With one-point control, the operator usually "over purifies" the uncontrolled composition. While this makes control relatively simple, it requires extra energy usage and reduces the capacity. However, there is an important case where one-point control often is optimal. This is when the column is operated at maximum capacity, e.g. maximum vapor load, and there effectively is only one degree of freedom left for composition control.

To select a good distillation control configuration, one should first consider the problem of controlling levels and pressure (y_2), and then consider the 2x2 composition control problem (y_1). Another important issue is that one often does not want large variations in the flows (L, V, V_T, D, B) because these changes usually cause disturbances in other parts of the plant. In particular, we often want to avoid large variations in D and B because these often are feed streams to downstream units.

For control applications, where conventional multiloop control systems are not satisfactory, an alternative approach, multivariable control, can be advantageous. In multivariable control, each manipulated variable is adjusted based on the measurements of all the controlled variables rather than on only a single controlled variable, as in multiloop control. The adjustments are based on a dynamic model of the process that indicates how the manipulated variables affect the controlled variables. Consequently, the performance of multivariable control, or any model-based control technique, will depend heavily on the accuracy of the process model. A specific type of multivariable control, model predictive control, has had a major impact on industrial practice.

4.4.2. Controller tuning

Once the type of feedback controller has been selected, the next step is to decide what values to use for its adjusted parameters. This is known as the controller tuning problem. There exist a large number of methods for controller tuning [16,17]. Most of them are based on heuristic rules that have been proven in practice, like the Ziegler-Nichols' rule [18]. The Ziegler-Nichols control settings are easy to find and to use and give reasonable performance on some loops, so they are benchmarks against which the performance of other controller settings is compared in many studies. This empirical method can be based on closed-loop testing (also called on-line tuning) of processes which are inherently stable, but where the system may become unstable, or it can be based on open-loop tests. In this work we will use the closed-loop test.

To implement the closed-loop Ziegler-Nichols method we use only proportional control in the experiment, so we set integral time to its maximum in the Simulink model (see Figure 3.6). We have two PI control loops, so we will use two configurations:

1. To study the control loop in x_B, we will open the loop in y_D
2. To study the control loop in y_D, we will open the loop in x_B

To open the loop without modifying the Simulink model we can set the proportional gain to 0. This will return no values for *deltaV* (or *deltaL*) variable, as if no control action was taken.

We just want to see the response of the control loop to a set point change, so we will disable feed changes in the model. Noise is also disabled.

The next step is to slowly increase the proportional gain until the system begins to exhibit sustained oscillations with a given small step set point. The proportional gain and period of oscillation at this point are the ultimate gain, K_u, and ultimate period, P_u. The Ziegler-Nichols settings are then computed from K_u and P_u by the formulas given in Table 4.1, depending on the type of controller.

Table 4.1.- Ziegler-Nichols settings for the three types of controllers (closed-loop method)

Parameter	Type of controller		
	P	PI	PID
K_c	$\frac{K_u}{2}$	$\frac{K_u}{2.2}$	$\frac{K_u}{1.7}$
τ_I	-	$\frac{P_u}{1.2}$	$\frac{P_u}{2}$
τ_D	-	-	$\frac{P_u}{8}$

Appendix IV - Ziegler-Nichols experiments show the results of the tests. For the x_B control loop we have that K_u = -103 and P_u = 5.8 min and for the y_D control loop K_u = 107 and P_u = 4.8 min (see Figure 4.9).

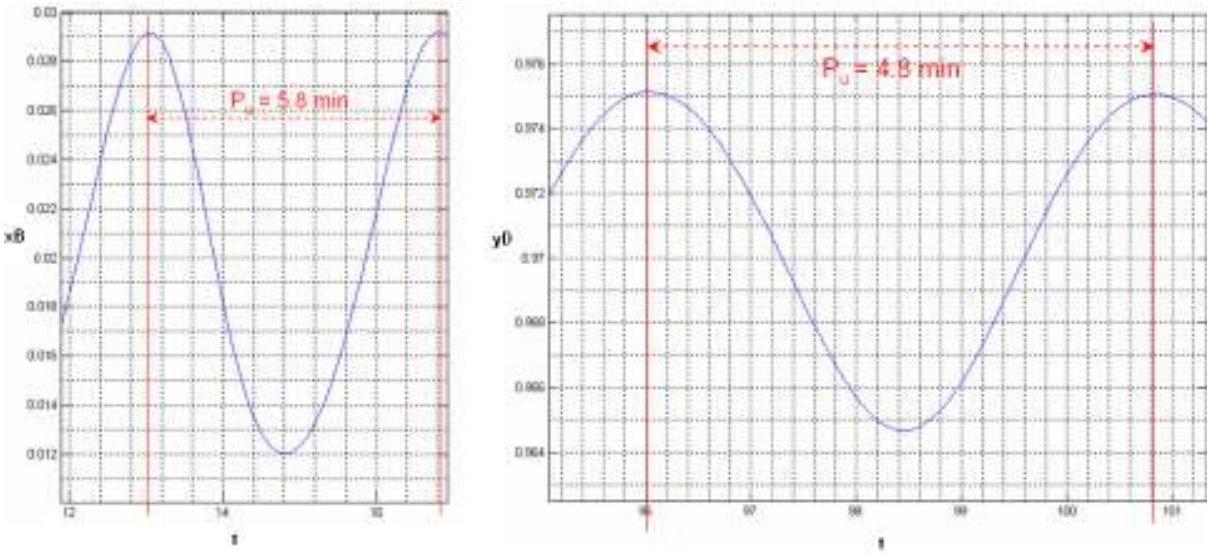


Figure 4.9.- Ultimate period for x_B and y_D control loops.

If we replace these values in the formulas of Table 4.1, we have the following control parameters:

Table 4.2.- Ziegler-Nichols tuning control parameters.

Parameter	Control loop	
	x_B	y_D
K_c	-46.82	48.64
τ_I	4.833	4.000

4.4.3. PI tuning optimization

Other tuning methods involve minimizing some cost function, usually based on time integral performance criteria such as the following:

1. *Integral Absolute Error* (IAE):

$$IAE = \int_0^\infty |e(t)|\, dt \qquad \text{Eq. 4.8}$$

The IAE simply integrates the absolute value and gives equal weight to large and small errors.

2. *Integral Squared Error* (ISE):

$$ISE = \int_0^\infty [e(t)]^2 dt \qquad \text{Eq. 4.9}$$

The ISE magnifies large errors. Thus minimization of this integral should help to suppress large, initial errors. The resulting controller setting tends to have a high proportional gain and the system is very under damped.

3. *Integral Time-weighted Absolute Error* (ITAE).

$$ITAE = \int_0^\infty t\, |e(t)|\, dt \qquad \text{Eq. 4.10}$$

The time weighting function gives a heavy penalty on errors that persist for long periods of time. This weighting function also helps to derive controller settings which allow for low settling times.

In all above equations, $e(t) = y(t) - r(t)$ where *y(t)* is the controlled variable and *r(t)* is the set point value, evaluated at time *t*.

4.4.3.1. Simulation time

The first factor to study is the simulation time that must be long enough to reflect the changes.

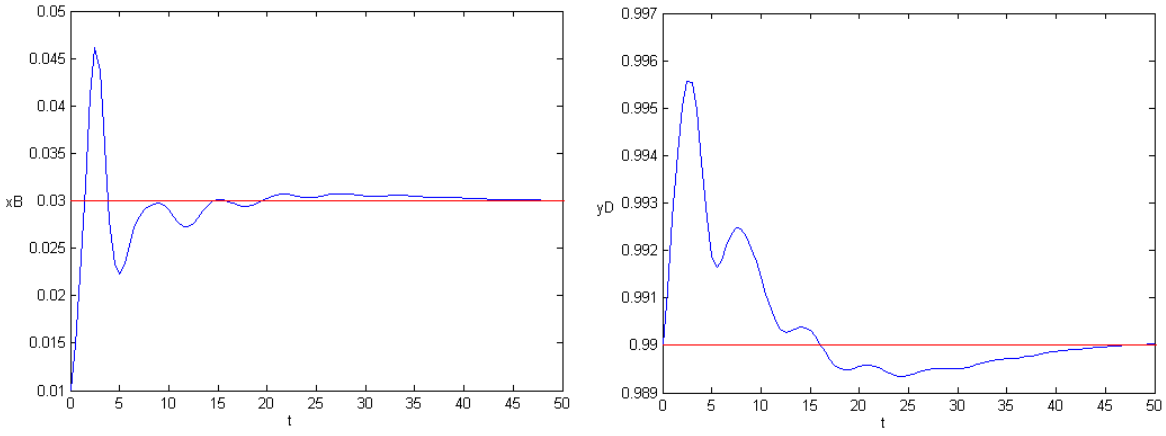


Figure 4.10.- Results for a set point change in x_B from 0.01 to 0.03

Figure 4.10 and Figure 4.11 show the results for a step change in x_B and in y_D (feed changes and noise have been deactivated).

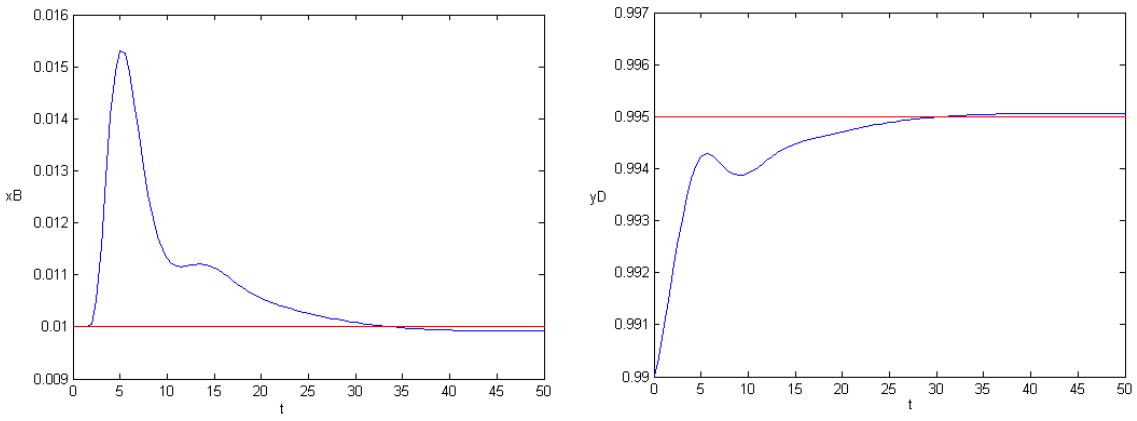


Figure 4.11.- Results for a set point change in y_D from 0.99 to 0.995

We suppose that steady state is reached at 50 minutes of simulation time. We compute the values of ISE, IAE and ITAE for the 50 min simulation and we use them as reference values. Then, we compare them with the values at another times (see Table 4.3 for results for the set point change in x_B and Table 4.4 for the results for the set point change in y_D). We can see that ISE and IAE variables reach more than 95% of total value with 40 min simulations. ITAE is time dependent so there is a great difference between 20 min and 50 min simulations, but with a 40 min simulation we reach more than 90% of total value (except for the case of $ITAE_{yD}$ = 87% of the set point change in y_D test, but it is still close to 90%). So we set 40 min as the simulation time for the tests.

Table 4.3.- Percentages reached at different times (set point change in x_B)

t (min)	ISE (xB)	IAE (xB)	ITAE (xB)	ISE (yD)	IAE (yD)	ITAE (yD)
20	99,2735	86,73083	45,77664	96,48805	79,32126	43,25879
30	99,72381	93,53237	68,7084	99,10652	92,01656	73,42018
40	99,97546	98,63315	92,14097	99,97056	98,95716	95,77046

The test for the optimization study is based on a set point change in x_B (or y_D). Under normal operating conditions (NOC), the distillation column suffers feed changes that provoke little changes around a given operating point, so there is a value for the controller performance variables (IAE, ISE or ITAE) under these conditions. Thus, we need a set point change that produces values for these performance variables that differ enough from the values under NOC. Otherwise, the ANOVA will not detect differences between control parameters changes during optimization.

Table 4.4.- Percentages reached at different times (set point change in y_D)

t (min)	ISE (xB)	IAE (xB)	ITAE (xB)	ISE (yD)	IAE (yD)	ITAE (yD)
20	99,13316	90,22189	72,59898	99,55305	91,95677	67,24649
30	99,92799	97,04213	88,10466	99,90333	96,34087	80,34384
40	99,94455	98,00441	91,38088	99,93329	97,70793	86,75898

The methodology followed for this sensitivity analysis is the following. We run five tests under NOC and we compute its corresponding performance variables. Then, we run other tests with different set point changes to check if there is enough difference between results. We start with the set point changes used previously for the determination of the simulation time: a set point change in x_B from 0.01 to 0.03 and a set point change in y_D from 0.99 to 0.995. The results are the following:

Table 4.5.- Results for a 40 min simulation under NOC for both control loops

#Test	ISE (x_B)	IAE (x_B)	ITAE (x_B)	ISE (y_D)	IAE (y_D)	ITAE (y_D)
1	$1{,}1846\cdot10^{-5}$	$2{,}2316\cdot10^{-2}$	$4{,}4003\cdot10^{-1}$	$3{,}9103\cdot10^{-6}$	$1{,}4170\cdot10^{-2}$	$3{,}2948\cdot10^{-1}$
2	$4{,}6720\cdot10^{-6}$	$1{,}4523\cdot10^{-2}$	$3{,}0940\cdot10^{-1}$	$2{,}3171\cdot10^{-6}$	$1{,}1047\cdot10^{-2}$	$2{,}6573\cdot10^{-1}$
3	$4{,}5980\cdot10^{-6}$	$1{,}6084\cdot10^{-2}$	$3{,}4876\cdot10^{-1}$	$1{,}1185\cdot10^{-6}$	$7{,}3774\cdot10^{-3}$	$1{,}6120\cdot10^{-1}$
4	$5{,}8434\cdot10^{-6}$	$1{,}7521\cdot10^{-2}$	$3{,}5630\cdot10^{-1}$	$9{,}9063\cdot10^{-6}$	$6{,}4556\cdot10^{-3}$	$1{,}5211\cdot10^{-1}$
5	$6{,}2953\cdot10^{-6}$	$1{,}7876\cdot10^{-2}$	$4{,}1490\cdot10^{-1}$	$7{,}7903\cdot10^{-6}$	$5{,}8473\cdot10^{-3}$	$1{,}2419\cdot10^{-1}$

Table 4.6.- Results for a set point change in x_B with a 40 min simulation (x_B control loop)

#Test	ISE (x_B)	IAE (x_B)	ITAE (x_B)
1	$1{,}6270\cdot10^{-3}$	$1{,}9593\cdot10^{-1}$	1,9230
2	$1{,}7749\cdot10^{-3}$	$2{,}0461\cdot10^{-1}$	1,8142
3	$1{,}6107\cdot10^{-3}$	$1{,}9701\cdot10^{-1}$	1,8876
4	$1{,}6224\cdot10^{-3}$	$1{,}8780\cdot10^{-1}$	1,8061
5	$1{,}5050\cdot10^{-3}$	$1{,}9123\cdot10^{-1}$	1,9355

Table 4.7.- Results for a set point change in y_D with a 40 min simulation (y_D control loop)

#Test	ISE (y_D)	IAE (y_D)	ITAE (y_D)
1	$1{,}1526\cdot10^{-4}$	$5{,}3166\cdot10^{-2}$	$4{,}1328\cdot10^{-1}$
2	$1{,}1805\cdot10^{-4}$	$5{,}7397\cdot10^{-2}$	$5{,}1756\cdot10^{-1}$
3	$1{,}1133\cdot10^{-4}$	$5{,}2661\cdot10^{-2}$	$4{,}6104\cdot10^{-1}$
4	$1{,}1827\cdot10^{-4}$	$5{,}5885\cdot10^{-2}$	$4{,}4397\cdot10^{-1}$
5	$1{,}2264\cdot10^{-4}$	$5{,}6728\cdot10^{-2}$	$4{,}3285\cdot10^{-1}$

We will use the test of the set point change in x_B to optimize the x_B control loop so we need to compare the results of the controller performance variables of this test with the ones under NOC. The same is valid for y_D. So we want to compare the following means:

Table 4.8.- Mean comparison scheme

Test: x_B set point change	ISE (x_B)	↔	ISE (x_B)	Test: normal operating conditions
	IAE (x_B)	↔	IAE (x_B)	
	ITAE (x_B)	↔	ITAE (x_B)	
Test: y_D set point change	ISE (y_D)	↔	ISE (y_D)	
	IAE (y_D)	↔	IAE (y_D)	
	ITAE (y_D)	↔	ITAE (y_D)	

The results obtained using the t-Student test for two mean comparison show that all the means are statistically different (p-Value<0.05) and, therefore, the proposed experiments are valid.

4.4.3.2. Sensitivity analysis

We will base the optimization methodology on DOE experiments, assessing the effects on response variables (IAE, ISE, ITAE) due to changes in factors (the controller parameters K_c and τ_I for both control loops). So we need to choose simulation parameters that lead to results that can detect such effects. Those parameters are the following:

- Simulation time

  In section 4.4.3.1 we have demonstrated that a 40 min simulation of a step change in both set points produces results in response variables that are different from the ones obtained under NOC. Now we want to know if we can use a shorter simulation time without losing the capability of detecting the effects in the response variables. We will test three simulation times: 20, 30 and 40 min

- Factor levels

  The factors of the experiments are the controller parameters K_c and τ_I for both control loops (x_B and y_D). The optimization methodology will be based on factorial designs at two levels, so we need to find values for the factors that give results that can be detected by the ANOVA of the design. These values will be obtained by applying a percentage of variation around a given central point. We need to find the minimum percentage of variation that yields detectable results. The optimization methodology will be based on linear relationships between variables, so we want to avoid high variations that could lead to nonlinearities in the response variables. The percentage of variations of the study will be 5, 10 and 15%

- Response variables

  These are the controller performance variables of each control loop (IAE, ISE, ITAE). The optimization methodology must use one of these response variables, so we need to choose the response variable that best detects the effects of the changes in the factors to study.

For this study we will use the experiments designed in section 4.4.3.2:

1. Set point change in x_B from 0.01 to 0.03
2. Set point change in y_D from 0.99 to 0.995

Factors will be studied at two levels that will be obtained by applying a given percentage variation around the Ziegler-Nichols parameter values (see section 4.4.2). For example, the factor values for a 5% variation will be the following (see Table 4.9).

Table 4.9.- Factors levels

x_B control loop

Factor	Var.	Low (-1)	High (+1)
A	K_c	-49.161	-44.479
B	τ_I	4.5914	5.0747

y_D control loop

Factor	Var.	Low (-1)	High (+1)
A	K_c	46.208	51.072
B	τ_I	3.8	4.2

We run a 2^2 factorial design with five replicated central points to search for curvature (see section 4.4.3.3). We apply this design for each combination of simulation time and percentage of variation of factors, and compute ANOVA to detect the effects in each response variable. The first set of experiments is done for a set point change in x_B from 0.01 to 0.03 and lets study the effects of K_c (factor A) and τ_I (factor B) of the control loop in x_B. The other set of experiments is based on a set point change in y_D from 0.99 to 0.995 to study the effects of K_c (factor A) and τ_I (factor B) of the control loop in y_D. Table 4.10 shows the results of these experiments. We can see whether the effects are statistically significant with p-Value<0.05 (1) or not (0). The studied effects are the simple effect of factor A and B, the interaction AB and the lack-of-fit test (see section 4.4.3.3). The response variables studied (ISE, IAE and ITAE) are computed from x_B for the experiments based on a set point change in x_B, and from y_D for the experiments based on a set point change in y_D.

We observe that response variables IAE and ITAE do not seem to be valid for the study, because they are not able to detect factor effects (cases colored in orange). ISE is the more appropriated response variable. A 20 min simulation time and a 5% variation appear to be enough to detect effects, except for the case of y_D control loop. In that case we see that with a 10% variation we can detect effects, so we are close to the zone that we are interested in. We will see that the methodology used for optimization (see section 4.4.3.3) starts with a low percentage of variation and increases it gradually until effects are detected. On the other hand,

a 15% variation produces the presence of curvature (red colored cases) that is something to be avoided at the initial steps of the optimization procedure.

Table 4.10.- Sensitivity analysis results

			ISE				IAE				ITAE			
Loop	**t_{Sim}**	**$\%_{Var}$**	**A_{eff}**	**B_{eff}**	**AB_{int}**	**Lack-of-fit**	**A_{eff}**	**B_{eff}**	**AB_{int}**	**Lack-of-fit**	**A_{eff}**	**B_{eff}**	**AB_{int}**	**Lack-of-fit**
x_B	20	5	1	0	0	0	0	0	0	0	0	0	0	0
		10	0	1	0	0	0	0	0	0	0	0	0	0
		15	1	1	0	1	0	0	0	1	0	0	0	1
	30	5	0	1	0	0	0	1	0	0	0	0	0	0
		10	0	1	0	0	0	0	0	0	0	0	0	0
		15	1	1	1	1	1	1	1	1	0	0	0	1
	40	5	0	0	0	0	0	1	0	1	0	1	0	1
		10	0	1	0	0	0	0	0	0	1	0	1	0
		15	1	1	1	1	0	0	0	0	0	0	0	0
y_D	20	5	0	0	0	0	0	0	0	0	0	0	0	0
		10	1	1	1	0	1	1	1	0	1	1	0	0
		15	1	1	0	0	1	1	0	0	1	1	0	0
	30	5	0	0	0	0	0	0	0	0	0	0	0	0
		10	1	1	1	1	1	1	0	0	0	0	0	0
		15	1	1	0	0	1	1	0	0	1	1	1	0
	40	5	0	0	0	0	0	0	0	0	0	0	0	0
		10	1	0	0	0	1	0	0	0	0	0	0	0
		15	1	1	0	1	1	1	0	0	0	0	0	0

*1: statistical differences (p-Value<0.05)

*0: no statistical differences

4.4.3.3. Control loop interaction

We have seen that the model has two control loops. One of them uses the manipulated variable V to control x_B and the other one uses L to control y_D. But, is there any interaction between both control loops? We want to know if modifying one of the manipulated variables affects not only its controlled variable but the other controlled one. The response variables to study are the controller performance variable ISE of both x_B and y_D controlled variables. The studied factors are the controller parameters of both control loops (Table 4.14). That is, we need to perform a 2^4 factorial design. The variation percentage to obtain the high and level factors is a 5% around the Ziegler-Nichols parameters. The experiments are based on a 20 min simulation of a set point change in x_B from 0.01 to 0.02 and y_D from 0.99 to 0.985. Results are shown in Table 4.11. These data are analyzed with Statgraphics Centurion XV. The first ANOVA analysis includes all the effects to see their relative importance. The pareto charts in Figure 4.12 show that for ISE_{xB} the main effects are A, B, C and AB and for ISE_{yD} these are C, D, B and A. In a second analysis the lower effects detected in the previous analysis are not considered anymore (Figure 4.13). In the case of x_B we can see that there is a full interaction between all controller parameters, not only the ones of x_B. It is surprising how the effect of the K_c of the y_D control loop has a greater

effect than the K_c of x_B control loop itself. The same situation occurs when analyzing the effects in y_D, where the controller parameters of the x_B control loop has more influence in y_D than its own control loop parameters. We must say that these interactions depend on the test used for the experiments: a positive step change in x_B will yield different results than a negative one. It also depends on the size of the step change, but, in any case, we can conclude that there is interaction between both control loops.

Table 4.11.- Results of the CCD design

A	B	C	D	ISE_{xB}	ISE_{yD}
-1	-1	-1	-1	0.0018170054	0.00019023400
-1	-1	-1	1	0.0018416740	0.00019225789
-1	-1	1	-1	0.0018991107	0.00018013591
-1	-1	1	1	0.0018661308	0.00018464276
-1	1	-1	-1	0.0016283518	0.00019069199
-1	1	-1	1	0.0016329068	0.00019254385
-1	1	1	-1	0.0016624461	0.00017820787
-1	1	1	1	0.0016626890	0.00018049129
1	-1	-1	-1	0.0015980105	0.00018768366
1	-1	-1	1	0.0015832749	0.00019248987
1	-1	1	-1	0.0016585320	0.00017926668
1	-1	1	1	0.0016602647	0.00018255927
1	1	-1	-1	0.0014936329	0.00018222561
1	1	-1	1	0.0014798407	0.00019082286
1	1	1	-1	0.0015440072	0.00017401687
1	1	1	1	0.0015259346	0.00017976038

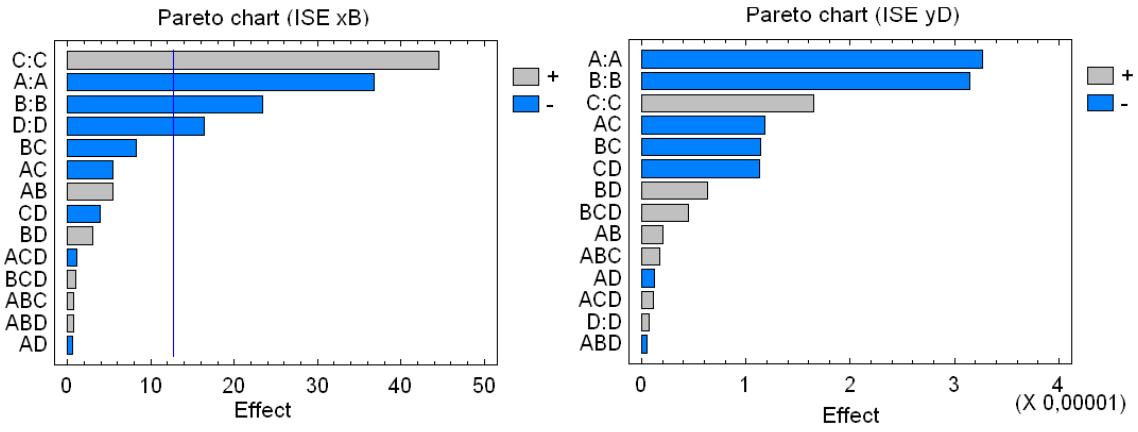


Figure 4.12.- Pareto chart for both response variables (until triple interaction)

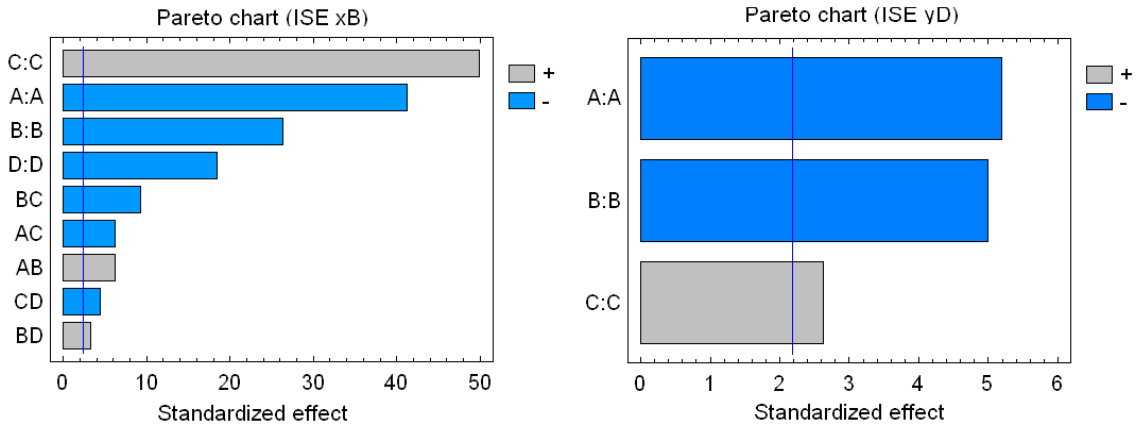


Figure 4.13.- Pareto chart for both response variables. Main effects.

4.4.3.4. Steepest ascent (descent) method

Now that we have designed the test to obtain the cost function, the following step is to design a methodology to search for optimum values.

Statistical techniques for optimization assume that a first-order model will serve as a good local approximation in a small region far away for the optimum [19]. Therefore, it makes sense to fit a simple first-order (or linear polynomial) model of the form:

$$Y = b_0 + b_1 X_1 + b_2 X_2 + \varepsilon \qquad \textbf{Eq. 4.11}$$

But near the optimum, fitting a first order model shows lack of fit (Figure 4.14). In that situation often a quadratic (second order) model suffices:

$$Y = b_0 + b_1 X_1 + b_2 X_2 + b_{12} X_1 X_2 + b_{11} X_1^2 + b_{22} X_2^2 + \varepsilon \qquad \textbf{Eq. 4.12}$$

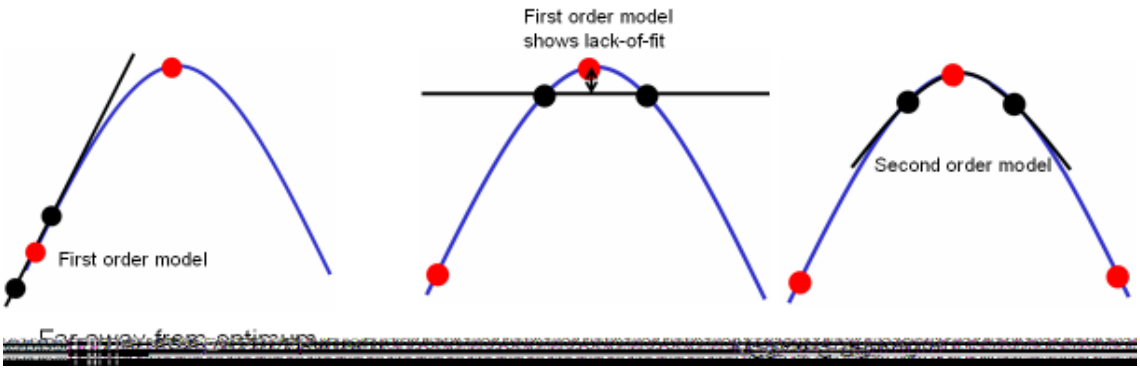


Figure 4.14.- Curvature effect near the optimum.

The path of steepest ascent (or descent) methodology is based on previous assumptions. It is divided into two phases (Figure 4.15):

1. In *Phase I* we determine the search direction (the steepest ascent, or descent, direction) and the length of the step to move from the current point. We keep experimenting along the direction until there is no further improvement in the response. At that point, a new fractional factorial experiment with centre runs is conducted to determine a new search direction. This process is repeated until at some point significant curvature in the response is detected, that implies that we are near an optimum. When significant curvature, or lack of fit, is detected, the experimenter should proceed with *Phase II*.
2. In *Phase II* a quadratic model is fitted to determine the optimum.

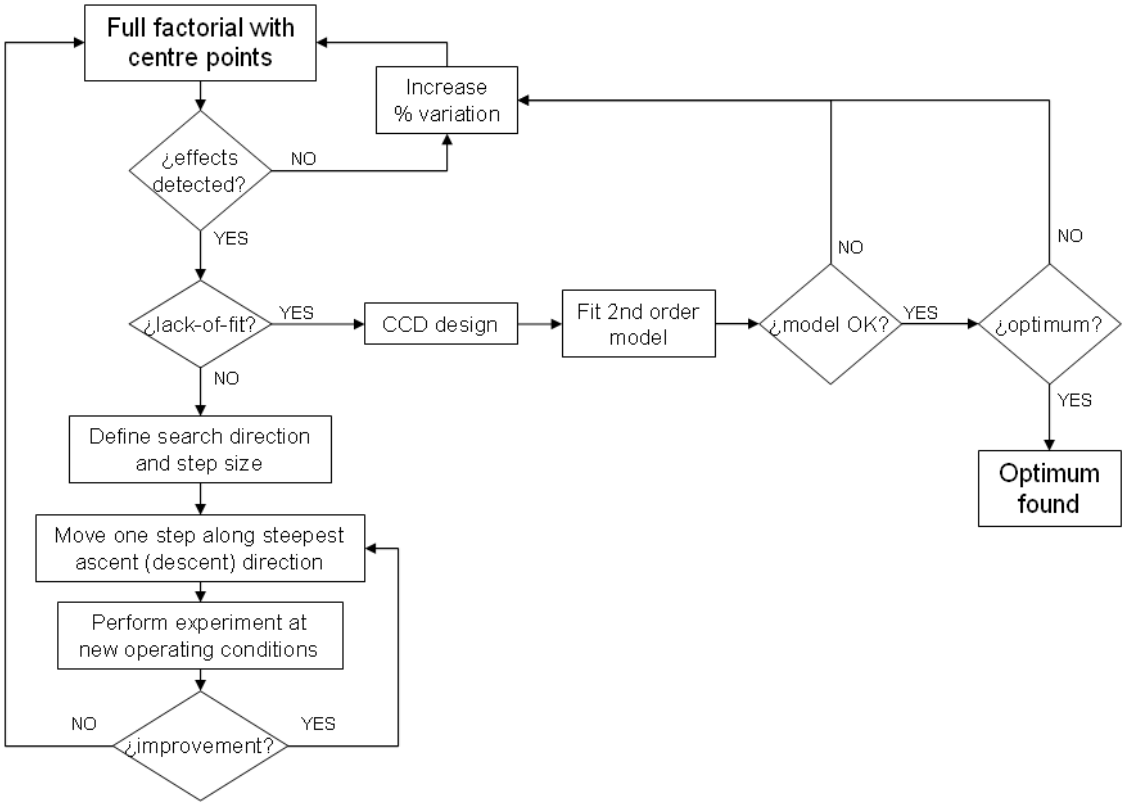


Figure 4.15.- Steepest ascent (descent) method

We optimize both control loops separately, so we have two factors (K_c and τ_I for each control loop) and one objective function that will be the controller performance variable ISE, that yields best results (see section 4.4.3.2). In previous sections we have demonstrated that the best model for the experiments is the following:

- Simulation time: 20 min
- Set point step change:
    i. From 0.01 to 0.03 for x_B
    ii. From 0.99 to 0.995 for y_D

That is, to find an optimum value for K_c and τ_I parameters for the x_B control loop we will use a 20 min simulation of a set point step change of x_B from 0.01 to 0.03. Factors will be the K_c and τ_I parameters of the x_B control loop, while K_c and τ_I parameters of the y_D control loop will not change. The objective function will be minimizing the ISE computed for the x_B variable.

On the other hand, to find an optimum value for K_c and τ_I parameters for the y_D control loop we will use a 20 min simulation of a set point step change of y_D from 0.99 to 0.995. Factors will be the K_c and τ_I parameters of the y_D control loop, while K_c and τ_I parameters of the x_B control loop will not change. The objective function will be minimizing the ISE computed for the y_D variable.

During *Phase I* a 2^2 factorial design with five centre points is run. Replicated centre points allow estimating possible curvature (see Eq. 14.1). The factor levels will be obtained by a percentage of variation around centre point. The initial percentage will be 5% and will be increased gradually a 5% if no effects are detected. The initial point will be the Ziegler-Nichols parameters (see Table 4.2).

The direction of steepest descent is determined by the gradient of the fitted model. The direction of the gradient, *g*, is given by the values of the parameter estimates, that is, $g' = (b_1, b_2, ..., b_k)$. The regression coefficients for the coded variables ($b_1, b_2, ..., b_k$) are one-half of the corresponding effect estimates. Since these parameter estimates $b_1, b_2, ..., b_k$ depend on the scaling convention for the factors, the steepest ascent (descent) direction is also scale dependent. That is, two experimenters using different scaling conventions will follow different paths for process improvement. This does not diminish the general validity of the method since the region of the search, as given by the signs of the parameter estimates, does not change with scale. An equal variance scaling convention, however, is recommended. The coded factors x_i, in terms of the factors in the original units of measurement, X_i, are obtained from the following relation:

$$x_i = \frac{X_i - \left(X_{low} + X_{high}\right)/2}{\left(X_{high} - X_{low}\right)/2}; \quad i = 1, 2, \ldots, k \qquad \textbf{Eq. 4.13}$$

This coding convention is recommended since it provides parameter estimates that are scale independent, generally leading to a more reliable search direction. The coordinates of the factor settings in the direction of steepest ascent, positioned a distance ρ from the origin, are given by:

$$\begin{aligned} &\text{minimize} \quad b_0 + b_1 x_1 + b_2 x_2 + \ldots + b_k x_k \\ &\text{subject to:} \quad \sum_{i=1}^{k} x_i^2 \le \rho^2 \end{aligned} \qquad \textbf{Eq. 4.14}$$

The solution is a simple equation that yields the coordinates:

$$x_i^* = \rho \frac{b_i}{\sqrt{\sum_{j=1}^{k} b_j^2}}; \quad i = 1, 2, \ldots, k$$ **Eq. 4.15**

An engineer can compute this equation for different increasing values of ρ and obtain different factor settings, all on the steepest descent direction. Then, we perform the experiment at these new operating conditions. We keep experimenting along the direction until there is no further improvement in the response.

During *Phase* II, we must vary the factors at 3 levels in order to fit a quadratic model. A 2^k design with centre points does not suffice, because then all quadratic factors are confounded. A 3^k design is possible, but not to be recommended because the number of runs grows fast and it is not efficient because it uses more runs than necessary to fit a quadratic model. More efficient designs are *Central Composite Designs* (*CCD*) or Box-Behnken designs (see -Appendix V - Adding secondary points-). Then, a second-order model is fitted using stepwise regression. If no quadratic model is obtained then the percentage of variation is increased and another factorial experiment is run. If a quadratic model is obtained, then we must determine the type of optimum. We can do this analytically through the matrix notation of the quadratic model:

$$\hat{Y}(x) = b_0 + b'x + x'Bx$$ **Eq. 4.16**

Where $b' = (b_1, b_2, \ldots, b_k)$ denotes a vector of first-order parameter estimates, $x' = (x_1, x_2, \ldots, x_k)$ is the vector of controllable factors and

$$B = \begin{pmatrix} b_{11} & b_{11}/2 & \cdots & b_{ik}/2 \\ & b_{22} & & \\ & & \ddots & \\ \text{symmetric} & & & b_{kk} \end{pmatrix}$$

is a matrix of second-order parameter estimates.

Equating the partial derivatives of $\hat{Y}$ with respect to x to zeroes and solving the resulting system of equations, the coordinates of the stationary point of the response are given by:

$$x^* = -\frac{1}{2} B^{-1} b$$ **Eq. 4.17**

The nature of the stationary point (whether it is a point of maximum response, minimum response or a saddle point, Figure 4.16) is determined by the matrix B. The two-factor interactions do not, in general, let us "see" what type of point x^* is. One thing that can be said is that if the diagonal elements of B (b_{ii}) have mixed signs, x^* is a saddle point. Otherwise, it is necessary to look at the characteristic roots or eigenvalues of B to see whether B is positive definite (so x^* is a point of minimum response) or negative definite (the case in which x^* is a point of maximum response).

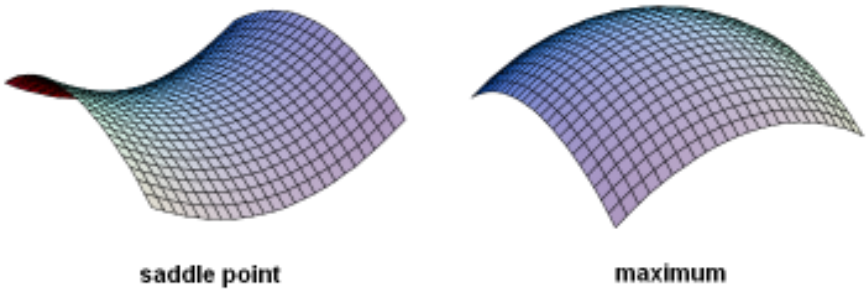


Figure 4.16.- Saddle point vs. maximum.

We have run several tests to find optimum values for controller parameters using the above methodology. The results for the x_B control loop optimization are shown in Table 4.12. We can see that the improvement depends on the number of tests needed to reach the optimum. The larger the number of test the longer the followed path and, in most cases, the better the results. Figure 4.17 compares the results of the best optimization (number 3 in Table 4.12) with the original Ziegler-Nichols parameters (see Table 4.2). Feed disturbances are avoided to see a more clear comparison (this provokes that results are not exactly the same that the ones obtained during optimization). We see that x_B is better controlled by the parameters found by optimization. We also have plotted the effect of the new x_B configurations parameters on the control of y_D (Figure 4.18). This is because there is an interaction between both controllers, but, as we can see, with a small effect.

Table 4.12.- Optimization results for x_B control loop

Id	Parameters		ISE (x_B)	Improv.	# of tests
	K_c	τ_I			
1	-39.730	5.8340	0.001400	10.22 %	58
2	-39.469	5.7694	0.001431	8.84 %	40
3	-43.450	8.6594	0.001189	23.55 %	256
4	-51.514	8.7648	0.001288	13.41 %	130
5	-36.305	6.4419	0.001306	13.62 %	103

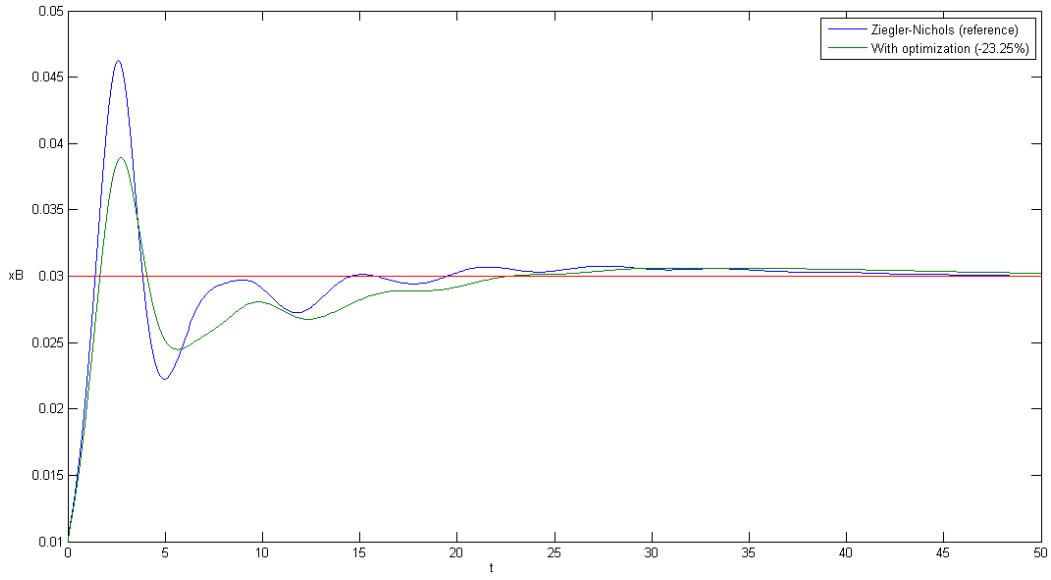


Figure 4.17.- Best results for the x_B control loop (x_B response)

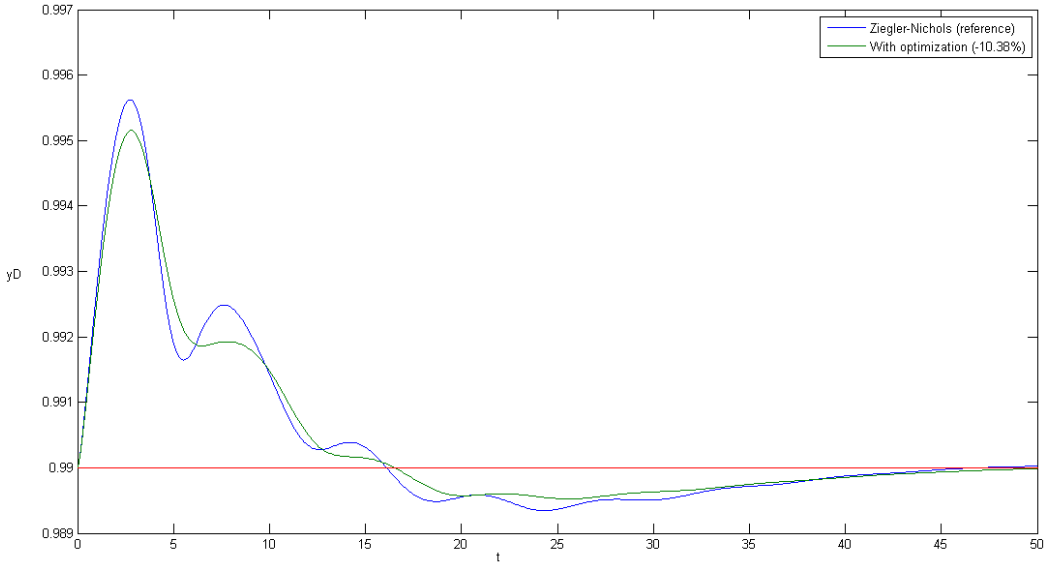


Figure 4.18.- Best results for the x_B control loop (y_D response)

Table 4.13 shows the results for the y_D control loop optimization. In that case improvement results are very similar. The best result is number 4 but, if we have into account the number of tests needed, optimization number 3 gets similar improvement results with less than half of tests.

Table 4.13.- Optimization results for y_D control loop

#	Parameters		ISE (y_D)	Improv.	# of tests
	K_c	τ_I			
1	151.49	6.2079	0.000074	37.58 %	166
2	108.94	3.7743	0.000074	38.65 %	112
3	91.112	2.9492	0.000072	38.42 %	76
4	101.32	3.4857	0.000072	39.00 %	157
5	102.57	3.4476	0.000073	37.29 %	148

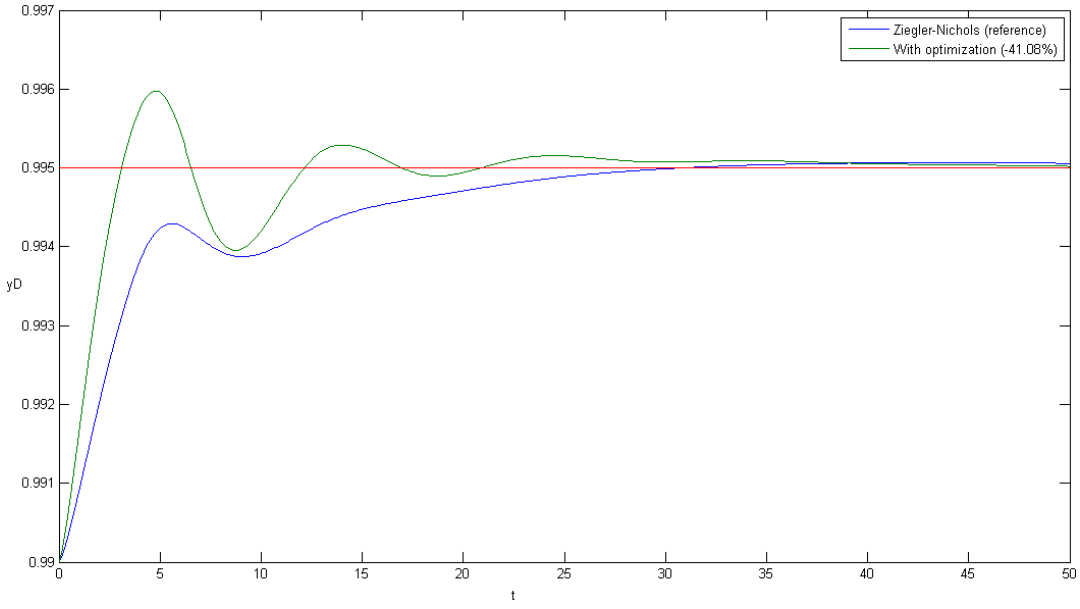


Figure 4.19.- Best results for the y_D control loop (y_D response)

Figure 4.19 compares the results of the best optimization (number 3 in Table 4.13) with the original Ziegler-Nichols parameters. We see that y_D is better controlled by the parameters found by optimization, but, in this case, the effect of the new configuration parameters of the y_D control loop on x_B variable is quite negative (Figure 4.20). In that case, the interaction between both controllers is high.

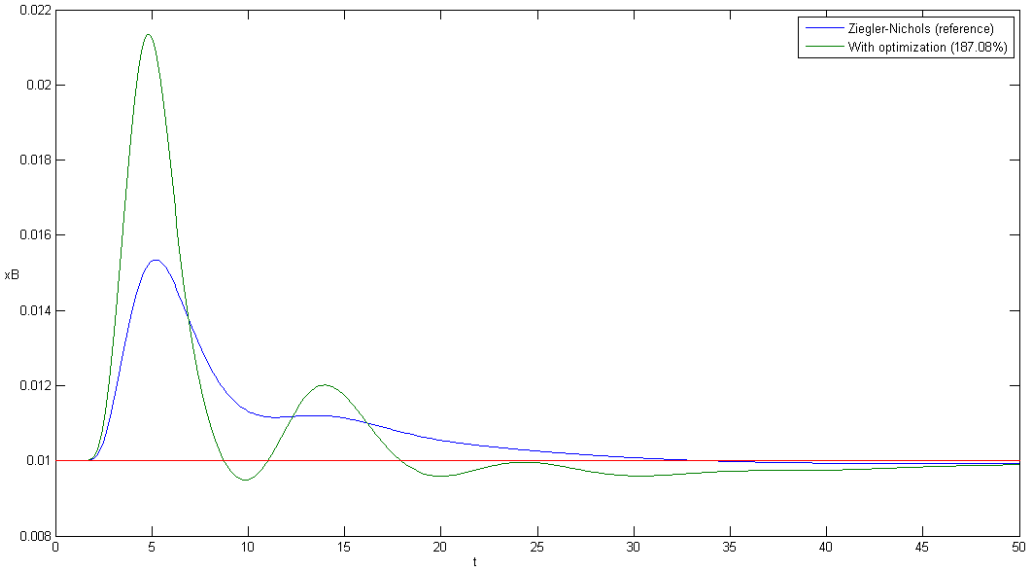


Figure 4.20.- Best results for the y_D control loop (x_B response)

4.4.3.5. Multiple responses

We have seen that there is interaction between both control loops and optimizing them separately is not the best solution. So we need an optimization method based on multiple responses. The optimization for the y_D control loop shown in Table 4.13 obtains a good improvement for ISE_{yD} (Figure 4.20) but diminishes the performance for the x_B control loop (Figure 4.19). In this situation, supposing that y_D is the primary response we can optimize mathematically the y_D control loop while keeping the x_B control loop performance under a minimum specification. We will use the experiment based on a set point change in x_B from 0.01 to 0.03 for x_B control loop optimization (Figure 4.10) and the experiment of a set point change y_D from 0.99 to 0.995 for y_D control loop optimization (Figure 4.11). We will study both responses ISE_{xB} and ISE_{yD} and four factors: the controller parameters K_c and τ_I of both control loops (Table 4.14).

Table 4.14.- Factors for the multiple response optimization

Factor	Variable	Control loop
A	K_c	x_B
B	τ_I	
C	K_c	y_D
D	τ_I	

The proposed methodology is the following:

1. Select an initial point for the optimization.
2. Set a percentage of variation to compute high and low factor levels.
3. Compute a central composite design (CCD) based on a 2^4 factorial design.
4. Fit a quadratic regression model using stepwise regression for both responses:

   $ISE_{yD} = f_1(A,B,C,D)$

   $ISE_{xB} = f_2(A,B,C,D)$

5. Perform a mathematical optimization using ISE_{yD} as primary function:

$$\begin{aligned} &\min\ ISE_{yD}(A,B,C,D) \\ &\text{subject to}: \quad ISE_{xB}(A,B,C,D) < ISE_{xB,\max} \\ &\qquad A_{\min} < A < A_{\max} \\ &\qquad B_{\min} < B < B_{\max} \\ &\qquad C_{\min} < C < C_{\max} \\ &\qquad D_{\min} < D < D_{\max} \end{aligned} \qquad \textbf{Eq. 4.18}$$

   That is, we minimize the performance of the y_D control loop while keeping the performance of the x_B control loop under a minimum that is the value of the performance for the initial parameters. The other constraints are the upper and lower bounds for the factors that are the star points in the CCD design. This is because the obtained models are only valid inside the studied zone defined the CCD design.

6. If a solution is obtained, run six simulations and compute their ISE_{yD} and ISE_{xB} for the new solution. Compare these values with the ones of the starting point. If both values improve with the new solution, set this new solution as the initial point for another study and proceed to step 2. If don't, stop optimization.

We will see the first step for the optimization of the y_D control loop as an example. The initial point will be the original Ziegler-Nichols parameters (see Table 4.2). The initial percentage of variation will be 20%, so the high and low values for the 2^4 factorial design are the following:

Table 4.15.- Factor levels for the CCD design.

Factor	Variable	Low	Center	High	$-\alpha$	$+\alpha$
A	K_c (x_B)	-56.184	-46.82	-37.456	-65.548	-28.092
B	τ_I (x_B)	3.8664	4.833	5.7996	2.8998	6.7662
C	K_c (y_D)	38.912	48.64	58.368	29.184	68.096
D	τ_I (y_D)	3.2	4	4.8	2.4	5.6

Table 4.16 shows the results of the CCD design. The values of the factors are the decoded ones.

Then, a stepwise regression is computed using *stepwisefit* MATLAB function. The fitted model is the following:

$$ISE = b_0 + b_1 A + b_2 B + b_3 C + b_4 D + b_{12} AB + b_{13} AC + b_{14} AD + b_{23} BC + b_{24} BD + b_{34} CD + b_{11} A^2 + b_{22} B^2 + b_{33} C^2 + b_{44} D^2 \quad \textbf{Eq. 4.19}$$

Table 4.16.- Results of the CCD design

A	B	C	D	ISE_{xB}	ISE_{yD}
-56.184	3.8664	38.912	3.2	0.0001208564	0.0001331156
-56.184	3.8664	38.912	4.8	0.0000867293	0.0001713909
-56.184	3.8664	58.368	3.2	0.0002100142	0.0000946589
-56.184	3.8664	58.368	4.8	0.0001554546	0.0001167078
-56.184	5.7996	38.912	3.2	0.0001714224	0.0001272632
-56.184	5.7996	38.912	4.8	0.0001226431	0.0001624012
-56.184	5.7996	58.368	3.2	0.0002648926	0.0000904576
-56.184	5.7996	58.368	4.8	0.0001973852	0.0001101010
-37.456	3.8664	38.912	3.2	0.0002378504	0.0001268424
-37.456	3.8664	38.912	4.8	0.0001669320	0.0001610492
-37.456	3.8664	58.368	3.2	0.0003840401	0.0000921472
-37.456	3.8664	58.368	4.8	0.0002764685	0.0001104359
-37.456	5.7996	38.912	3.2	0.0003391319	0.0001207837
-37.456	5.7996	38.912	4.8	0.0002407394	0.0001509133
-37.456	5.7996	58.368	3.2	0.0004958955	0.0000882281
-37.456	5.7996	58.368	4.8	0.0003644565	0.0001036359
-46.82	4.833	48.64	4	0.0002112365	0.0001172280
-46.82	4.833	48.64	4	0.0002112402	0.0001172288
-46.82	4.833	48.64	4	0.0002111665	0.0001172120
-46.82	4.833	48.64	4	0.0002111773	0.0001172220
-46.82	4.833	48.64	4	0.0002111801	0.0001172220
-46.82	4.833	48.64	4	0.0002112024	0.0001172149
-65.548	4.833	48.64	4	0.0001240802	0.0001228881
-28.092	4.833	48.64	4	0.0004728796	0.0001102634
-46.82	2.8998	48.64	4	0.0001524811	0.0001266068
-46.82	6.7662	48.64	4	0.0002753493	0.0001126194
-46.82	4.833	29.184	4	0.0001179193	0.0001898311
-46.82	4.833	68.096	4	0.0003175549	0.0000886067
-46.82	4.833	48.64	2.4	0.0003287871	0.0000933652
-46.82	4.833	48.64	5.6	0.0001646301	0.0001424703

We compare the forward model with the backward one. In MATLAB, the criterion used for adding or removing variables is based on p-Value. A p-Value = 0.05 is used in both cases. The models comparison is based on the R^2 value computed as:

$$R^2 = 1 - \frac{SS_{resid}}{SS_{total}} \quad \textbf{Eq. 4.20}$$

where SS_{resid} is the sum of squares of the residuals and SS_{total} is the total sum of squares of the response. These values are obtained from the *stats* structure returned by the *stepwisefit* function.

These are the results of stepwise regression for values in Table 4.16:

Table 4.17.- R^2 for stepwise regression.

Response	Mode selection	R^2
ISE_{xB}	Forward	0.8988
ISE_{xB}	Backward	0.9949
ISE_{yD}	Forward	0.9937
ISE_{yD}	Backward	0.9947

In that case, results are similar for both models of ISE_{xB}, although backward model is slightly better. For ISE_{yD}, the backward model is selected. These models are shown in Table 4.18.

Table 4.18.- Stepwise regression models.

Coefficient	ISE_{xB}	ISE_{yD}
b_0	4.89425E-004	2.03147E-004
b_1	2.41047E-005	-
b_2	1.22135E-004	-
b_3	1.56501E-005	-5.45561E-006
b_4	-1.55389E-004	3.61265E-005
b_{12}	1.32332E-006	-
b_{13}	1.44160E-007	8.23511E-009
b_{14}	-1.69656E-006	-1.85005E-007
b_{23}	-	-
b_{24}	-6.38039E-006	-8.71636E-007
b_{34}	-8.74251E-007	-5.00816E-007
b_{11}	2.39093E-007	-
b_{22}	-	-
b_{33}	-	5.64289E-008
b_{44}	1.25342E-005	-

Next step is a mathematical nonlinear optimization where our primary function is ISE_{yD} and the constraint function is ISE_{xB} (see Eq. 4.18). The functions and lower and upper bounds are the following:

> ISE_{yD} *(A,B,C,D)*: quadratic model obtained from the substitution of ISE_{yD} coefficients of Table 4.18 in Eq. 4.19.
>
> ISE_{xB} *(A,B,C,D)*: quadratic model obtained from the substitution of ISE_{xB} coefficients of Table 4.18 in Eq. 4.19.
>
> $ISE_{xB,max}$ =0.0002112005, is the mean value of ISE_{xB} at the center point.

The lower and upper bounds for the factors are the minimum and maximum values in the CCD design in Table 4.16, that is:

Table 4.19.- Lower and upper bound for factors constraints.

	A	B	C	D
min	-65.548	2.8998	29.184	2.4
max	-28.092	6.7662	68.096	5.6

To perform this optimization we use *fmincon* MATLAB function using interior-point algorithm that has been found the best for our situation.

In our example, the function *fmincon* founds a local minimum. The results of the optimization are the following:

ISE_{yD} = 0.0000859188

A = -65.53 (K_c for x_B control loop)

B = 4.303 (τ_I for x_B control loop)

C = 54.54 (K_c for y_D control loop)

D = 2.438 (τ_I for y_D control loop)

Next step is to check if the performance variables at new point are really better than at the initial point. Results for both performance variables are shown in Table 4.20. We perform a one-tailed t-test for the two samples of ISE_{xB} and another one for the two samples of ISE_{yD} at significance level $\alpha = 0.05$. The null hypothesis is that both means are equal and the alternative hypothesis that the mean for the initial point is smaller than the mean for the new point:

H_0: $mean_{initial} = mean_{new}$

H_1: $mean_{initial} < mean_{new}$

Table 4.20.- ISE values for initial and new points.

Initial point		New point	
ISE_{yD}	**ISE_{xB}**	**ISE_{yD}**	**ISE_{xB}**
0.0001172280	0.0002112365	0.0000891766	0.0002020435
0.0001172288	0.0002112402	0.0000891756	0.0002020587
0.0001172120	0.0002111665	0.0000891725	0.0002020216
0.0001172220	0.0002111773	0.0000891736	0.0002020319
0.0001172220	0.0002111801	0.0000891749	0.0002020772
0.0001172149	0.0002112024	0.0000891775	0.0002021160

We use the *ttest2* MATLAB function with parameter *tail = 'left'*. If the function returns 1 indicates a rejection of the null hypothesis at the 5% significance level, that is, the mean for the new point is greater than the mean for the initial point, so no improvement is reached. The comparison is done for both ISE_{xB} and ISE_{yD}. If any of these variables increases, then the optimization process stops.

In this case, both performance variables for the new point are smaller than the ones for the initial point, so improvement is reached:

ISE_{yD} improvement = -23.93%

ISE_{xB} improvement = -4.345%

We use the solution found as the center point for the new CCD design and the iterative process starts over again. Step 2 yields the following results:

R^2_{xB} = 0.994645; R^2_{yD} = 0.988320
Kp_xB = -65.530872
Ki_xB = 4.297720
Kp_yD = 54.539223
Ki_yD = 2.461615
ISE_yD = = 0.000089 (-22.676787%)
ISE_xB = = 0.000200 (-5.270528%)

This time the solution does not improve the results in step 1, so the iteration process finishes and the final solution is the one obtained in first iteration step.

If we compare results between the original Ziegler-Nichols parameters (see Table 4.2) and the ones obtained by optimization (Figure 4.21) we see that we have improved the performance of the y_D control loop without penalizing the performance of the x_B control loop which is a better result that the one obtained by single response optimization (see section 4.4.3.3).

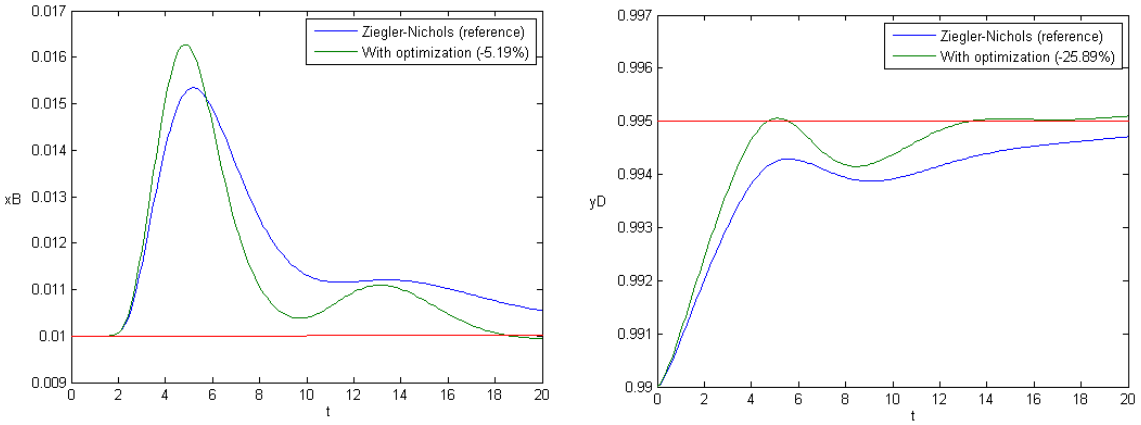


Figure 4.21.- Comparison of results after optimization (change in y_D set point).

The main problem is that this solution is still very under damped and, in some situations may cause instability. The optimization is based on a step change in y_D, so, if we use these results for the experiment based on a step change in x_B, we can see that results are completely unacceptable (see Figure 4.22). In fact, due to the interaction between both control loops, it is almost impossible to find a solution that works fine in any situation and a compromise solution is necessary.

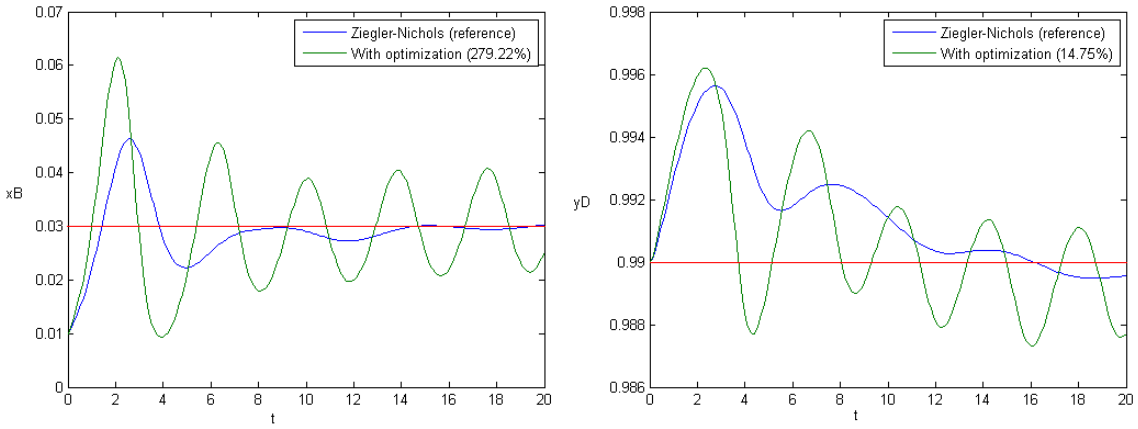


Figure 4.22.- Comparison of results after optimization (change in x_B set point).

We have seen an example for the optimization of the y_D control loop as the primary function using the performance in the x_B control loop as a constraint. Here, the experiment is based on a step change of 0.005 in y_D. On the other hand, when our primary goal is to optimize the x_B control loop the objective function is ISE_{xB} and the constraint is based on ISE_{yD}. In this case, the simulation is based on a step change of 0.02 in x_B (Figure 4.10). Results are shown in Table 4.21 (extended results are shown in Appendix VI). The column "Primary function" indicates the primary objective function used during the optimization process.

Table 4.21.- Results for the percentage of variation study.

Primary function	Variation percent.	Step	% Improvement	
			ISE_{xB}	ISE_{yD}
ISE_{xB}	5	1	-12.62	-0.4079
	10	-	-	-
	15	1	-17.28	-1.031
	20	1	-13.71	-17.28
	25	1	-17.17	-15.74
	30	2	-18.06	-18.52
ISE_{yD}	5	3	-3.572	-15.59
	10	2	-0.8964	-16.26
	15	2	-0.8880	-14.84
	20	1	-4.345	-23.93
	25	1	-5.257	-13.27
	30	-	-	-

We can see in the extended results (see Appendix VI) that the larger the percentage of variation the smaller the R^2 of the fitted model, although it is still acceptable. On the other hand, when the variation is too high process become unstable and simulation fails (this occurs when the percentage of variation is greater than 30%).

We can observe that with this method we use fewer tests than for the steepest descent method. This method is based on a CCD design with a total of 30 observations and one design is used in every optimization step. If an optimization process stops at step 3, then we have needed 90 simulations. This is because, unlike the steepest descent method, now we do not need to start with a linear model and we fit directly a quadratic model. That lets us to start with high percentage of variations of factor levels, that is, to explore a greater zone and to obtain results with fewer iterations. Best results are obtained for variations between 15 and 30%. The best result for ISE_{xB} is for a 30% variation and the best result for ISE_{yD} is for a 20% variation (see Table 4.21), that are the results of the above example. Table 4.22 shows the values of the controller parameters for these results.

Table 4.22.- Best results for both optimization processes.

Parameter	ISE_{xB} optimization	ISE_{yD} optimization
K_c (x_B)	-43.31	-65.53
τ_I (x_B)	8.524	4.303
K_c (y_D)	56.74	54.54
τ_I (y_D)	4.158	2.438

Figure 4.23 and Figure 4.24 show the results for both optimizations. We have seen that the results for the optimization of the y_D control loop are valid when applied for the simulation used for optimization (the step change in y_D, in this case), but they are not appropriated when applied to the step change in x_B. Something similar occurs for x_B: the results for the optimization of the x_B control loop improve the performance of both controllers when applied to the set point change in x_B (Figure 4.23) but they are not appropriate for the set point change in y_D (Figure 4.24). Again, it is difficult to find a solution valid in all cases.

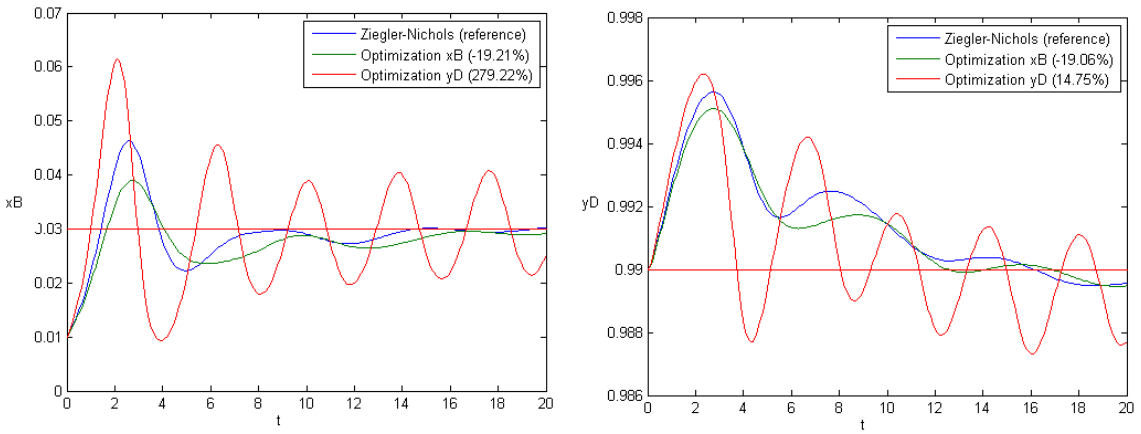


Figure 4.23.- Comparison of both optimizations (change in x_B set point).

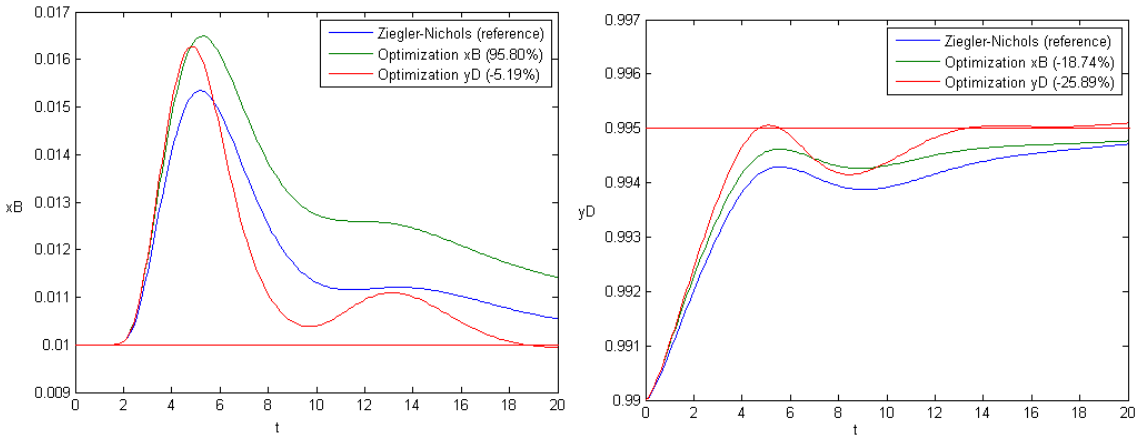


Figure 4.24.- Comparison of both optimizations (change in y_D set point).

5. Troubleshooting

5.1. Introduction

Close to 300 case histories of malfunctioning columns were extracted from the literature and summarized in Table 5.1, classified according to their principal causes [20].

Table 5.1.- Causes of column malfunctions.

Cause	Number of reported cases	Percent of reported cases
Instrument and control problems	52	18
Troublesome column internals	51	17
Startup and/or shutdown difficulties	48	16
Operational difficulties	38	13
Reboilers, condensers	28	9
Primary design (VLE, column size, …)	21	7
Foaming	18	6
Installation mishaps	16	6
Tray and downcomer layout	13	4
Relief problems	12	4
Total	**297**	**100**

An analysis of Table 5.1 suggests the following:

- Instrument and control problems, startup and/or shutdown difficulties, and malfunctioning column internals are the major single causes of column malfunctions. Among them, they make up more than half of the reported incidents.
- Reboilers, condensers and operation difficulties amount to about half of the remaining problems. Thus three out of four incidents are caused by either these or the factors previously mentioned. Familiarity with these problems, therefore, is of great importance to persons involved in distillation and absorption operation and troubleshooting.
- Primary design problems, foaming, installation mishaps, relief problems and tray and downcomer layout problems make up the rest of the column malfunctions. Familiarity with these problems is useful to troubleshooters and operation personnel, but only one incident out of every four is likely to be caused by one of these factors.
- Primary design is an extremely wide topic, encompassing vapor-liquid equilibrium, reflux-stages relationship, stage-to-stage calculations, unique features of multicomponent distillation, tray and packing efficiencies, scale-up, column diameter determination, flow patterns, type of tray, and size and material of packing. This topic

occupies the bulk of most distillation texts and perhaps represents the bulk of our present distillation know-how.

This work will be focused on operational problems.

5.2. Major disturbances

The type and magnitude of disturbances affecting a distillation column have a direct effect on the resulting product variability. These are the major types of disturbances encountered in distillation columns [(20)]:

1. Feed composition upsets

Changes in the feed composition represent the most significant upsets with which a distillation control system must deal on a continuous basis. A feed composition change shifts the composition profile through the column resulting in a large upset in the product compositions. Most industrial columns do not have a feed composition analyzer; therefore, feed composition upsets usually appear as unmeasured disturbances. When a feed composition analyzer is available, a feed forward controller can be applied using the on-line measurements of the feed composition (see Figure 4.2). Because feed composition changes represent a major disturbance for distillation control, the sensitivity of potential control configurations to feed composition upsets is a major issue for configuration selection.

2. Feed flow rate upsets

The flow rates in a steady-state model of a column with constant tray efficiencies scale directly with column feed rate. Therefore, ratio control (using L/F, D/F, V/F or *B*/F as composition controller output) is an effective means of handling feed flow rate upsets. Dynamic compensation is normally required to account for the dynamic mismatch between the response of the product compositions to feed flow rate changes and the response to changes in the MVs. When certain ratios (e.g., L/D, V/*B*) are used as MVs, these ratios, combined with the level control, automatically compensate for feed flow rate changes.

3. Feed enthalpy upsets

For columns that use a low reflux ratio, feed enthalpy changes can significantly alter the vapor/liquid rates inside the column, causing a major shift in the internal composition profile and, therefore, a significant upset in the product compositions. This upset can be difficult to identify because most industrial columns do not have feed temperature measurements and even if a feed temperature measurement is available, it does not detect feed enthalpy changes for a two-phase feed. This disturbance may be difficult to distinguish from feed composition upsets without a more detailed analysis. It may be necessary to install a feed preheater or cooler to maintain a constant feed enthalpy to a column.

4. Subcooled reflux changes

When a thundershower passes over a plant, the reflux temperatures for the columns can drop sharply. Columns that use finned-fan coolers as overhead condensers are particularly susceptible to rapid changes in ambient conditions. If internal reflux control is not applied, severe upsets in the operation of the columns result because of major shifts in the composition profiles of the columns. When internal reflux control is correctly applied, the impact of a thunderstorm on column operations can be effectively eliminated.

5. Loss of reboiler steam pressure

When a steep drop in steam header pressure occurs, certain columns (those operating with control valves on the reboiler steam that are nearly fully open) experience a sharp drop in reboiler duty. This results in a sharp increase in the impurity levels in the products. When the steam header pressure returns to its normal level, the composition control system for the column attempts to return to the normal product purities. Because of the severity of this upset, if the composition controllers are not properly tuned, the upset can be amplified by the composition controllers, requiring the operators to take these controllers off-line to stabilize the column, greatly extending the duration of the period of production of off-specification products. This disturbance is, in general, the most severe disturbance that a control system on a distillation column must handle and may require invoking overrides that gradually bring the operation of the column to its normal operating window instead of expecting the composition controllers to handle this severe upset by themselves.

6. Column pressure upsets

Column pressure has a direct effect on the relative volatility of the key components in the column. Thus, changes in the column pressure can significantly affect product compositions. A properly implemented pressure control scheme maintains column pressure close to its set point, with only short-term and low-amplitude departures. A large class of columns (e.g., refinery columns) is operated at maximum condenser duty to maximize column separation, which minimizes steam usage. For these cases, the column pressure increases during the day, when the cooling water or ambient air temperature is the greatest, and decreases at night, but the resulting pressure changes are usually slow enough that the composition controller can efficiently reject this disturbance.

5.3. Regulatory controls

Improperly functioning flow, level or pressure controllers can undermine the effectiveness of the product composition controllers.

5.3.1. Flow controllers

Flow controllers are used to control the flow rates of the products, the reflux and the heating medium used in the reboiler and their set points are determined by the various level and composition or temperature controllers. To assess the performance of a flow control loop, you can apply block sine waves and comparing these results for the dead band and time constant with the expected performance levels.

5.3.2. Level controllers

Level controllers are used to maintain the level in the accumulator, the reboiler and the intermediate accumulator of a stacked column (*i*.e., a distillation column composed of two separate columns when there are too many trays for one column). Loose level control on the accumulator and reboiler has been shown to worsen the composition control problem for material balance control configurations (when either *D* or *B* is used as a *MV* for composition control). When *D* or *B* is adjusted, the internal vapor/liquid traffic changes only after the corresponding level controller acts as a result of the change in *D* or *B*. On the other hand, if a level controller is tuned too aggressively, it can result in oscillations passed back to the column and contribute to erratic operation. When the reboiler duty is set by the level controller on the reboiler, a level controller that causes oscillation in the reboiler can also cause cycling in the column pressure.

6. Multivariate statistical process control

6.1. Introduction

Statistical Process Control (*SPC*) concepts and methods have become very important in the manufacturing and process industries [(21)]. 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 totally different from 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 event and removing them (rather than simply continuing to compensate for them), the process is improved.

Conventional *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 [(22)]

$$T^2 = (z-\mu)^T S^{-1} (z-\mu) \qquad \text{Eq. 6.1}$$

where **z** is a (*K* x 1) vector of measurements; μ is the in-control (*K* x 1) mean vector; and **S** is an estimate of the in-control (*K* x *K*) covariance matrix Σ. This approach assumes that $z \sim N_K(\mu, \Sigma)$ and checks if the mean of the process 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. 6.1, *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 [23] (*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 [24]. 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* [25] (Partial Least Squares) can also be used.

6.2. *MSPC* based on *PCA* (*MSPC-PCA*)

The *MSPC-PCA* 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 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 [26].

6.2.1. Phase I. Model building

The main goal in Phase I is to model the in-control process performance based on a set of historical in control (reference) data. 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 as

commented below. 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. 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

$$X = \sum_{a=1}^{A} t_a p_a^T + \sum_{a=A+1}^{rank(X)} t_a p_a^T = TP^T + E = X^* + E \qquad \textbf{Eq. 6.2}$$

P (KxA) is the loading matrix containing the loading vectors 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 t_a of the score matrix *T* ($t_a = X\ p_a$) 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, $X^* = TP^T$. Matrix *E (NxK)* contains the residuals (statistical noise), *i*.e. the information that is not explained by the *PCA* model.

Eq. 6.2 shows that the *PCA* model transforms each K-dimensional original observation vector x_i (*ith* row of matrix *X*) into an *A*-dimensional score vector $t_i^T = \{t_{i1}, t_{i2}, ...t_{iA}\}$ (*ith* row of matrix *T*) and a residual vector e_i (*ith* 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 [(27)] and to stop based on different criteria [(23,28)]. 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 [(29)].

6.2.1.1. Model size

The main purpose of Phase I is to obtain a model that is able to detect abnormal situations and to accomplish this it is based on observations under normal operating conditions (NOC). The simulation time is proportional to the number of observations for a given sample time. We could think that 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 optimal number of observations that renders a model able to detect abnormal situations with a minimum computational effort. To find this number we will run tests with several simulation times and compute the percentage of points that are outside control limits. To detect if a point is in control with respect to the correlation of the model we use SPE statistic (see section 6.2.1.3).

The methodology to determine the optimal number of observations is based on selecting a random model and testing it against the whole dataset. The size of the random model increases gradually till it reaches an acceptable percentage of points out of control. The iterative process follows these steps (Figure 6.1):

1. Select a random subset of observations from the dataset. The number of observations *n* in this subset is a percentage *P* of the total number of observations n_T in the full dataset. This percentage starts with a 0.1% and increases by 0.1% in each iteration.
2. Compute the PCA model for the selected subset of observations.
3. Exploit the obtained model with the full set and compute percentage of points out of control in SPE control chart.
4. Repeat steps 1 through 3 several times to get a mean.
5. If the mean of percentage of out of control points is less than the allowable percentage then the process stops. To do this, we perform a one-tailed test for the mean at significance level 0.1. The allowable percentage of points out of control is $100 \times \alpha$. The model used for PCA monitoring will be the best model in the group. Otherwise, we proceed to step 1.

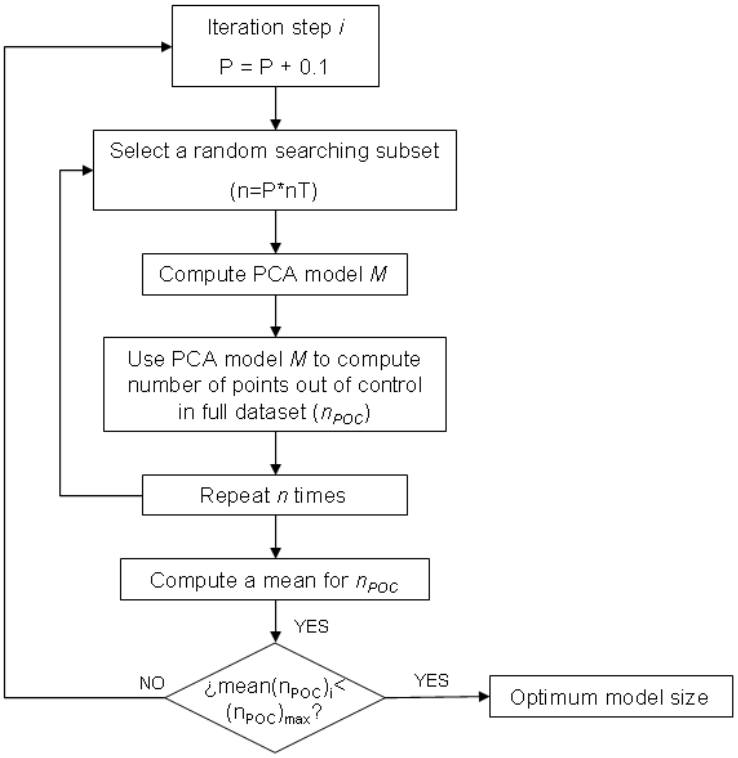


Figure 6.1.- Methodology for the optimal number of observations.

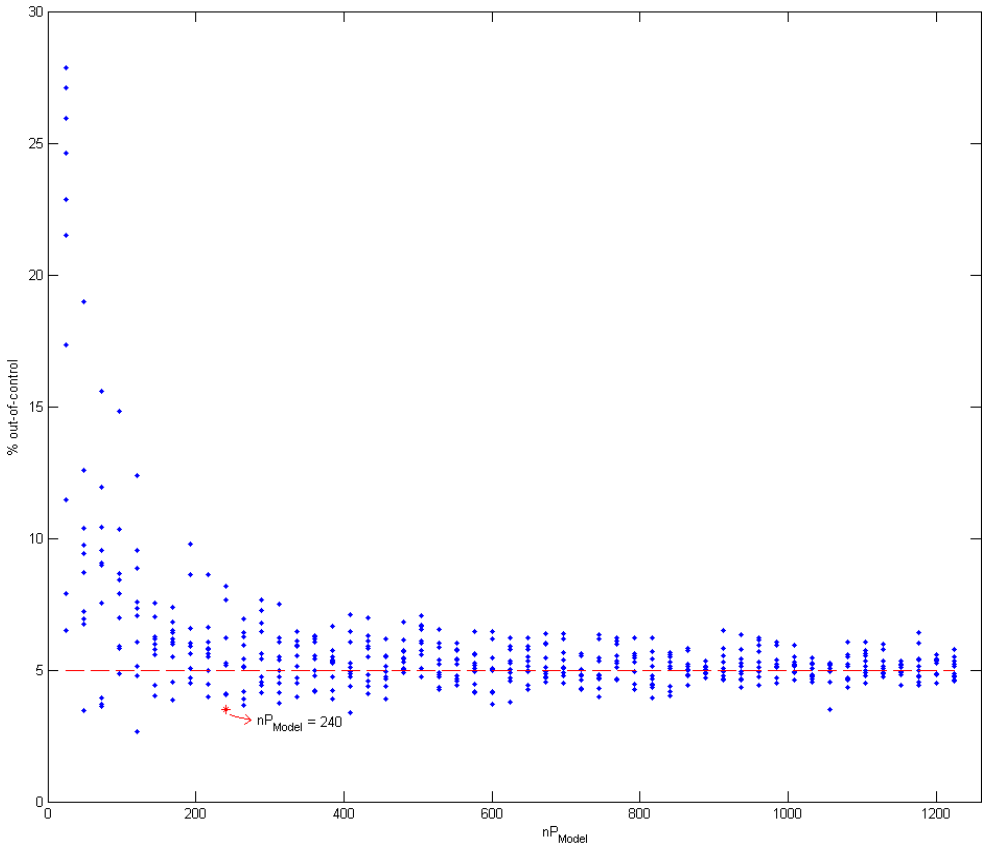


Figure 6.2.- Points out of control vs. model size (SPE)

We have used this optimization process with a set of 24000 observations obtained from a 12000 min simulation with a sample time of 0.5 min. We can see how the number of points out of control in SPE chart decreases rapidly with model size (Figure 6.2). In this case a model of 240 observations (120 min) renders 3.49% percentage of points out of control when applied over the full 24000 observations dataset.

Figure 6.3 shows the mean and 95% confidence intervals of the percentage of points out of control. We can observe how it converges towards 5% while model size increases. We get acceptable models when model size is greater than approximately 200 observations, which means a simulation time of 100 min (for a 0.5 min sample time).

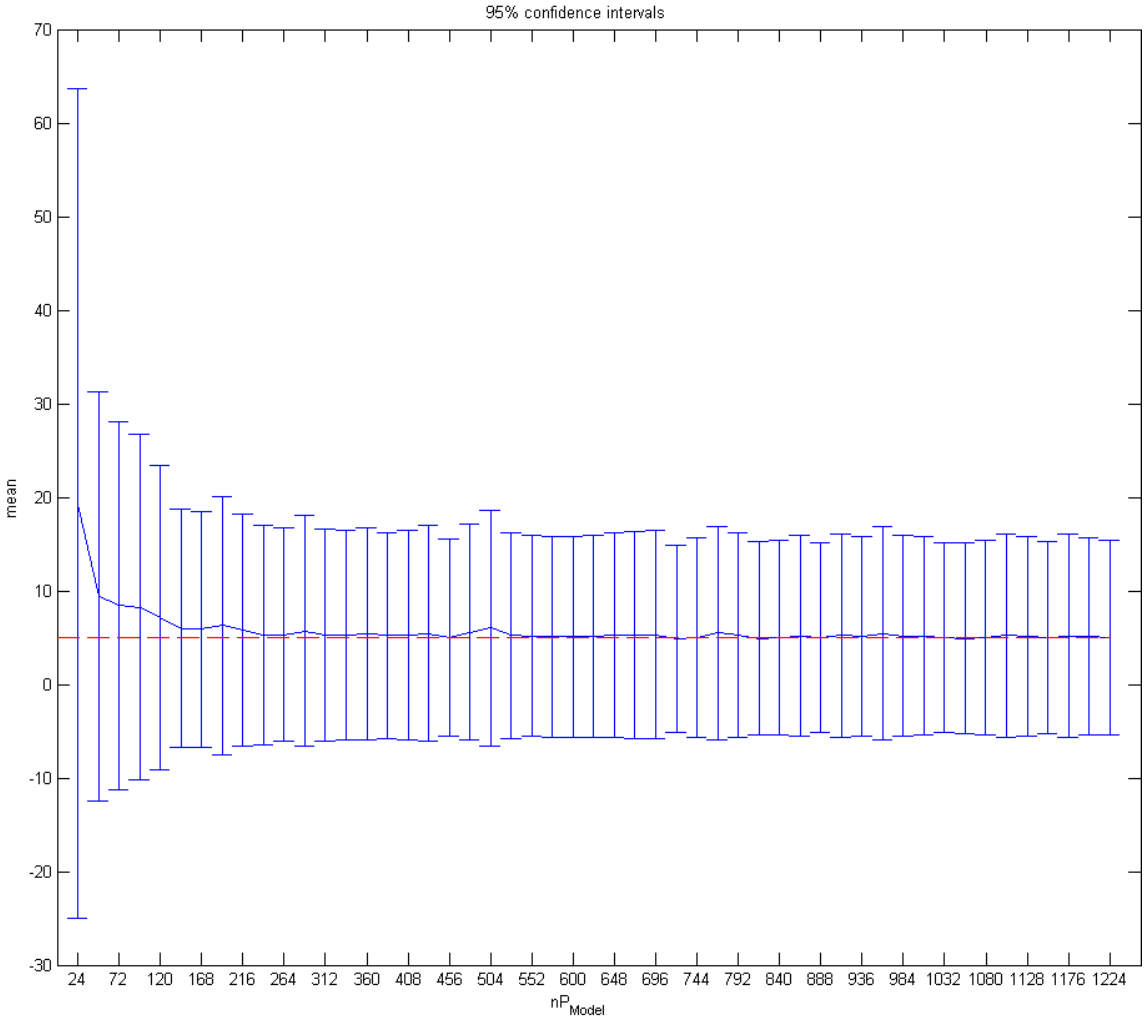


Figure 6.3.- Percentage mean vs. model size

6.2.1.2. *PCA* with MATLAB

As an example we will use a model selected above (see section 6.2.1.1). The number of observations is 240 that represent a 120 min simulation with a sample time of 0.5 min. Then, the matrix data for model validation (*Z* matrix) has 240 observations (*N*=240). The selected variables are F_V, z_F, T_F, *L*, *V*, *D*, *B* and the 41 tray temperatures (T_1,…,T_{41}), so *K*=48.

Table 6.1.- MATLAB code for *PCA*

```
% Z – original data matrix (N observations x K variables)
% X – standardized data matrix (N x K)
X = zscore(Z);
% P – loading matrix (K x K)
% T – score matrix (K x K)
% L – latent variables (K-dimension vector)
[P,T,L] = princomp(X);
```

By using the Statistics Toolbox of MATLAB, we can use the *princomp* function to compute the *PCA* (see Table 6.1). The results of the *princomp* function 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.

We also can compute *PCA* through Singular Value Decomposition (*SVD*),

Table 6.2.- 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 confidence region for a two dimensional score plot of dimension *a* and *b* is an ellipse with axis [30]:

$$\pm\sqrt{s_{ta(b)}^2 \times F_{2,N-2}^{1-\alpha} \times 2 \times \frac{N^2-1}{N\left(N-2\right)}} \qquad \textbf{Eq. 6.3}$$

Usually, the statistical significance level α is 0.05.

$s^2_{ta\,(b)}$ 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*:

$$s^2_{ta} = \mathrm{var}(t_i) = \lambda_i \qquad \textbf{Eq. 6.4}$$

, where λ_i represents the *i-th*-element of vector *L*.

In a 2-D plot (that is, A=2), the equation of the ellipse that limits the confidence region will be the following (see red ellipse in Figure 6.4):

$$\left(\frac{t_1}{a}\right)^2 + \left(\frac{t_2}{b}\right)^2 = 1 \qquad \textbf{Eq. 6.5}$$

Table 6.3.- MATLAB code for the scattered plot of the *PCA* scores.

```
% 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);
a = sqrt(L(1)*F*2*(N^2-1)/(N*(N-2)));
b = sqrt(L(2)*F*2*(N^2-1)/(N*(N-2)));
rectangle('Position',[-axis1,-axis2,2*axis1,2*axis2],'Curvature',[1,1], ...
          'EdgeColor','r');
```

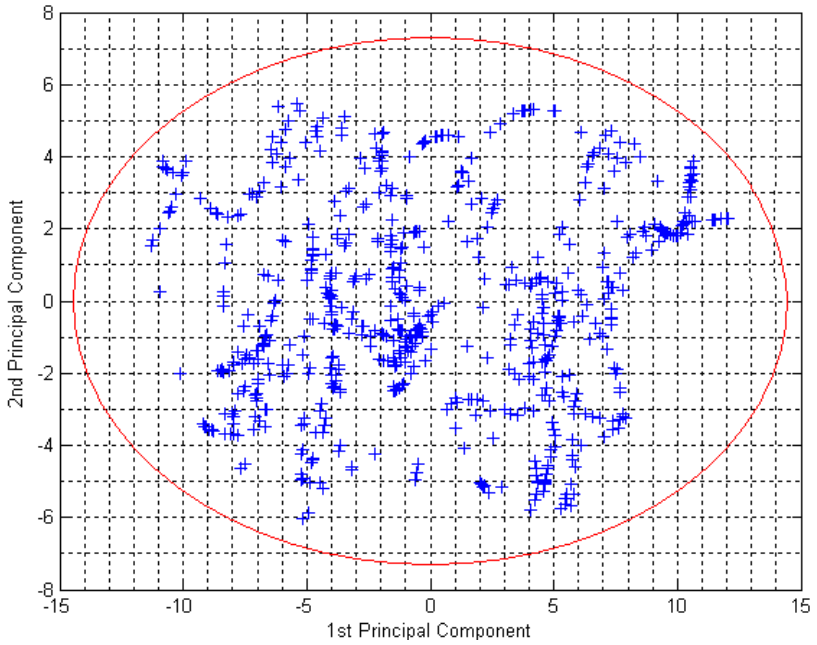


Figure 6.4.- Scattered plot of the scores in 2-D latent subspace.

Figure 6.4 shows a plot of the original observations projected onto the latent subspace formed by the first two principal components (see MATLAB code in Table 6.3). Actually, the number of components extracted in a principal component analysis is equal to the number of observed variables being analyzed. 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 significant *PCs* [30]. In our case we will retain the first *A PCs* that accumulate more than 90% of explained variance (see Table 6.4).

Table 6.4.- 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 95.48% of variance explained is reached by the third *PC*, so *A*=3. Figure 6.5 shows a pareto chart with the percentage of variance explained by each *PC*.

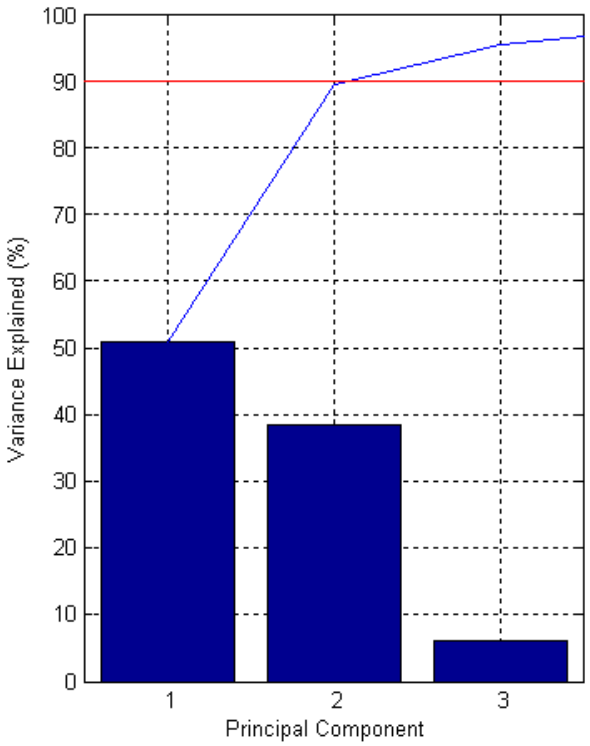


Figure 6.5.- Variance explained by the first principal components.

According to Eq. 6.2, we can compute the residuals matrix *E* in the following way (note that we only get the first *A* elements in both loading and score matrices):

```
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;
```

6.2.1.3. *PCA* statistics

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

$$T_A^2 = t_i^T \Theta^{-1} t_i = \sum_{a=1}^{A} \frac{t_a^2}{\lambda_a} \qquad \textbf{Eq. 6.6}$$

where $\Theta(A \times A)$ is the covariance matrix of *T* (diagonal matrix of the highest *A* eigenvalues $\{\lambda_1,...,\lambda_A\}$). This statistic is the *Hotelling-T²* statistic when a reduced subspace with *A* components is used instead of the original variables space, and it represents the estimated Mahalanobis distance from the center of the latent subspace to the projection of an observation onto this subspace. Table 6.5 shows the MATLAB code that computes *Tsquared*, a vector of dimension N that contains T_A^2 for each observation in matrix *X*.

Table 6.5.- MATLAB code for *Hotelling-T²* 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 significance level 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;
```

Under the assumption that the scores follow a multivariate normal distribution (they are linear combinations of random variables), it holds [31] that in Phase I, T_A^2 (times a constant) follows a beta (*B*) distribution

$$T_A^2 \sim \frac{(N-1)^2}{N} B_{A/2,(N-A-1)/2} \qquad \textbf{Eq. 6.7}$$

while in Phase II, T_A^2 (times a constant) follows an F distribution

$$T_A^2 \sim \frac{A(N^2-1)}{N(N-A)} F_{A,(N-A)} \qquad \textbf{Eq. 6.8}$$

The difference in both distributions 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. 6.8 can also be used for approximating the distribution of the T_A^2 statistic in Phase I.

On the other hand, the SPE statistic for ith observation x_i is given by

$$SPE = e_i^T e_i = \left(x_i - x_i^*\right)^T \left(x_i - x_i^*\right) \qquad \textbf{Eq. 6.9}$$

, where e_i is the residual vector of *i-th* observation, and x_i^* is the prediction of the observation vector 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 6.6.- MATLAB code for *SPE* statistic computation

```
% Sum of squared prediction errors
SPE = diag(E()*E()');
% UCL for SPE at significance level 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);
```

6.2.1.4. Multivariate control charts

From the above two statistics, in *MSPC-PCA* two complementary multivariate control charts are constructed. 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. 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 [32]. 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.

Upper control limits (UCL) for the Shewhart T_A^2 chart at significance level (type I risk) α can be obtained for Phase I from Eq. 6.7

$$UCL\left(T_A^2\right) = \frac{(N-1)^2}{N} B_{(A/2,(N-A-1)/2),\alpha} \qquad \textbf{Eq. 6.10}$$

where $B_{(A/2,(N-A-1)/2),\alpha}$ is the 100(1-α)% percentile of the corresponding beta distribution that can be computed from the 100(1-α)% percentile of the corresponding *F* distribution by using the relationship [31]

$$B_{(A/2,(N-A-1)/2),\alpha} = \frac{\left(A/\left(N-A-1\right)\right)F_{(A,N-A-1),\alpha}}{\left(1+\left(A/\left(N-A-1\right)\right)F_{(A,N-A-1),\alpha}\right)} \qquad \textbf{Eq. 6.11}$$

For Phase II, the corresponding UCL from Eq. 6.8 is given by

$$UCL\left(T_A^2\right)_\alpha = \frac{A\left(N^2-1\right)}{N\left(N-A\right)} F_{(A,(N-A)),\alpha} \qquad \textbf{Eq. 6.12}$$

Regarding the UCL for the Shewhart SPE chart, several procedures can be used. In this work we will use an approximation based on the weighted chi-squared distribution ($g\chi_h^2$) proposed by Box [33]. Nomikos and MacGregor [34] 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 significance level α is given by

$$UCL(SPE)_{\alpha} = \frac{v}{2b}\chi^2_{(2b^2/v),\alpha} \qquad \textbf{Eq. 6.13}$$

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

6.2.2. Phase II. Model exploitation

6.2.2.1. 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 z_i is available, after pre-processing it is projected onto the *PCA* model yielding the scores and the residuals, from which the value of the *Hotelling's* T_A^2 and the value of the *SPE* 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 T_A^2 and *SPE* 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 *MSPC-PCA*, one of the most widely used approaches is the contribution plots approach [(32)]. 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.

6.2.2.2. 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

$$Cont(SPE; x_{new,k}) = sign\left(e_{new,k}\right) e^2_{new,k} = sign\left(e_{new,k}\right)\left(x_{new,k} - x^*_{new,k}\right)^2 \qquad \textbf{Eq. 6.14}$$

Then, variables with high contributions are investigated.

6.2.2.3. 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 $\left(t_{new,a}/\lambda_a\right)^2$ is plotted and the *ath* score with the highest normalized value is selected.
2) The contribution of each original *kth* variable to this *ath* score at this new abnormal observation is given by

$$Cont\left(t_{new,a};x_{new,k}\right)=p_{ak}\,x_{new,k} \qquad \textbf{Eq. 6.15}$$

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

A plot of these contributions is created. Variables on this plot with high contributions but with the same sign as the score should be investigated (contributions of the opposite sign, will only make the score smaller). When there are some scores with high normalized values, an overall average contribution per variable can be calculated, over all the selected scores [(24)].

7. Results

A Graphical User Interface (GUI) is developed to interact with the Simulink model shown in 3.2 and to implement the Multivariate Statistical Process Control strategy shown in 6.2. This GUI is developed using the tool GUIDE of MATLAB R2009b. Figure 7.1 shows the main window, with a tree view control that lets navigate through the benchmark. Every menu option has its corresponding button in the integrated toolbar that gives more operability. The window shows basic information about the benchmark: its name and a simple description updatable by the user. Below this, there are two forms, depending on the phase (Phase I or Phase II).

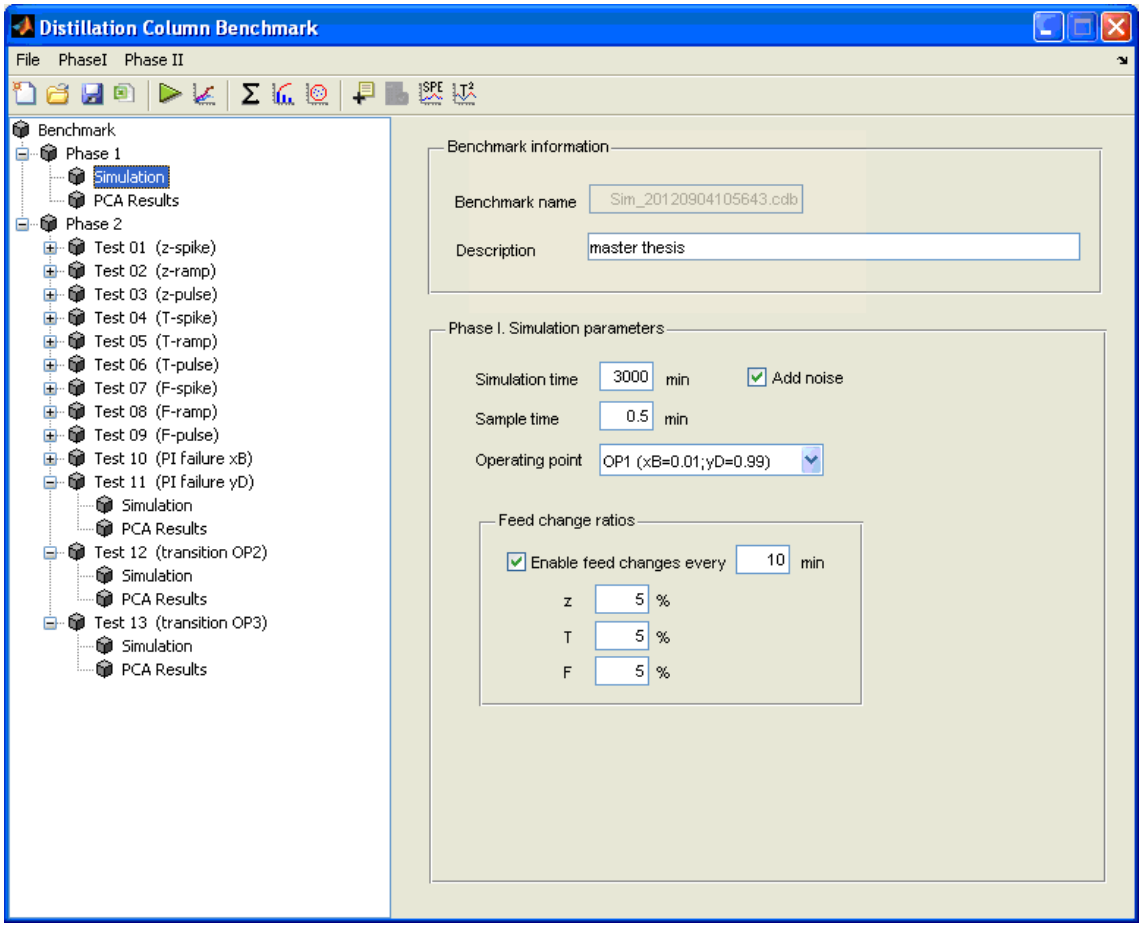


Figure 7.1.- Main window. Phase I simulation parameters.

Figure 7.2 shows the application menu and toolbar that summarizes all the functionality. The menu is divided into three main groups. The first one takes into account options related to file management, the second one deals with options used to develop a PCA model during Phase I and the last one 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. The application works with structures that are saved to a mat-file with the extension ".cdb" ("column distillation benchmark").

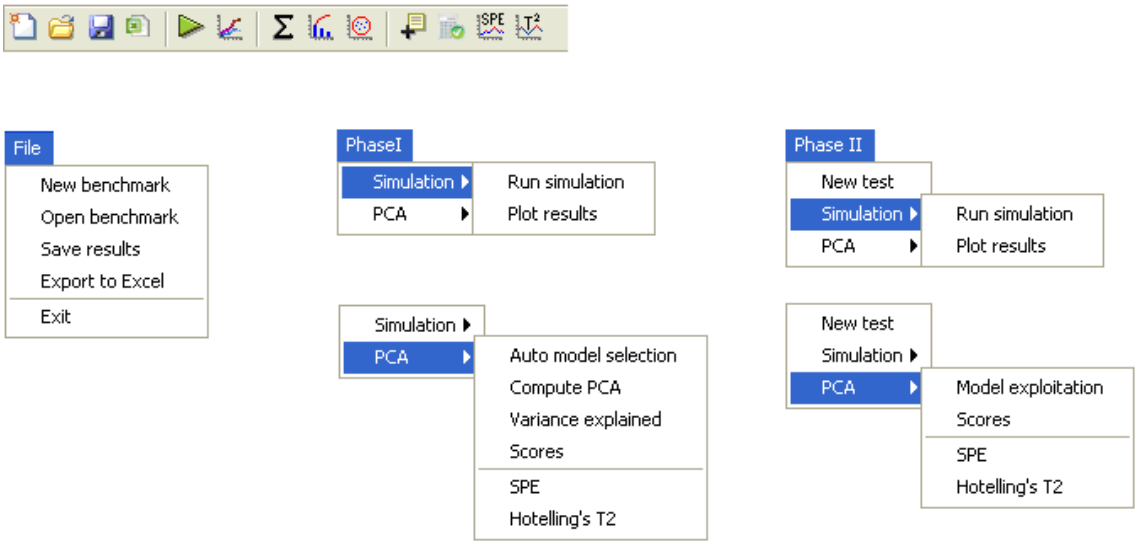


Figure 7.2.- Application menu and toolbar.

User can save its progress at any time as well as export simulation results to an Excel file. Each tab in this Excel file corresponds to a test in the benchmark and it stores the simulation parameters of the test and its results.

7.1. Phase I. Model development

7.1.1. Simulation parameters

Figure 7.1 displays the window shown when selecting the “Simulation” or “PCA Results” nodes of “Phase I” in the tree view, and contains information about the parameters used for the simulation of the model under normal operating conditions (phase I). This data will be used to render the PCA model for later analysis.

The parameters that are available for the simulation are the following:

1. Simulation time
2. Add noise: if checked, this will add white noise to the data as explained in 3.2.5
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 as shown in 2.2.5
5. Enable feed changes:
    a. “Every ... min” text box: indicates the time between changes
    b. Change ratios (z, T, F): percentage used for the changes in the input signals of the model (composition, temperature and flow of the feed).

The benchmark generated for our example is the following:

Table 7.1.- Phase I parameters for benchmark#1

Parameter	Value
Simulation time	12000 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 = 5%, F = 5%, T = 5%

When the simulation is set up, the user can run it. Then the Simulink model opens (if it was closed), the parameters are updated and the simulation starts. When the simulation finishes, results are stored and plotted in a graph (Figure 7.3). The user can access to this plot again using the "Plot results" menu option or the corresponding toolbar button.

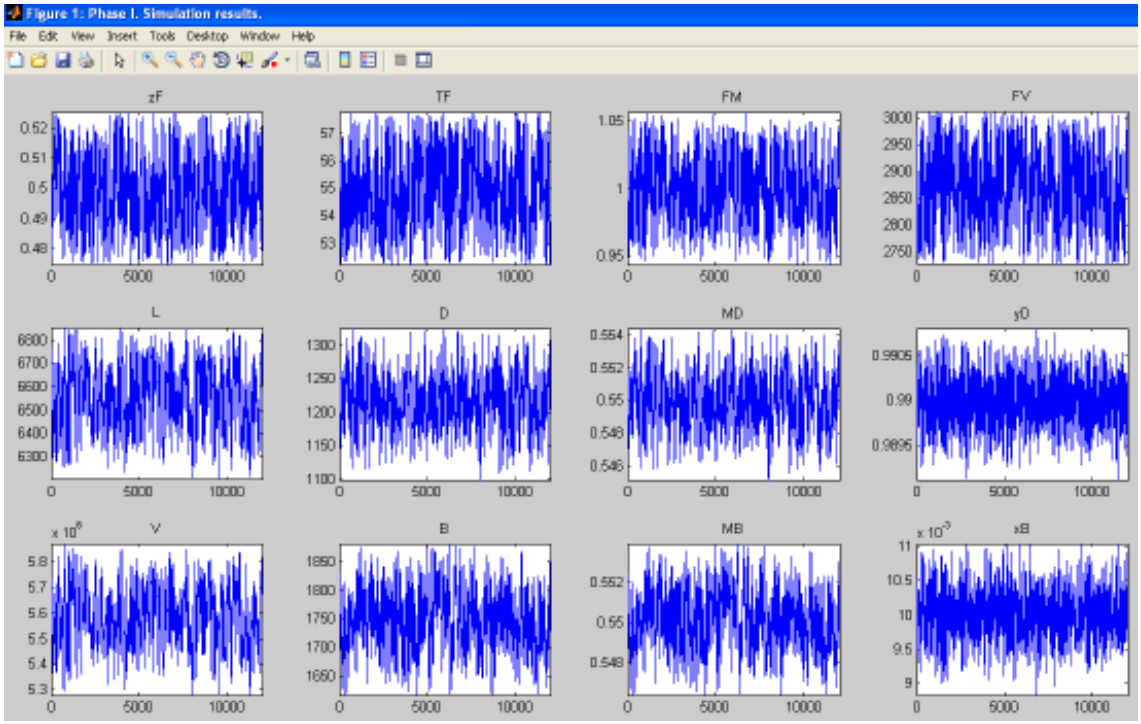


Figure 7.3.- Phase I simulation results.

The results plotted are the following:

zF, feed composition
TF, feed temperature (ºC)
FM, feed molar flow (kmol/min)
FV, feed volumetric flow (L/h)
L, reflux flow (L/h)
D, distillate flow (L/h)
MD, condenser holdup (kmol)
yD, distillate composition
V, boilup flow (L/h)
B, bottoms flow (L/h)
MB, reboiler holdup (kmol)
xB, bottoms composition

7.1.2. PCA computation

Once the simulation is done, user can use the compute a PCA model for the simulation results that will be used later for monitoring. Here, user has two options:

1. Search for a model with minimum size that is able for monitoring of the full simulated data as explained in 6.2.1.1. This corresponds to the “Auto model selection” menu option.
2. Construct a full model based in all data. This corresponds to the “Compute PCA” menu option.

The first option is based on the methodology presented in section 6.2.1.1 and it is useful to get a minimum model size (that is, minimum computing requirements) without losing monitoring capability.

In any case, PCA is computed through SVD as explained in 6.2.1.2. When the computation finishes two graphs are shown:

1. The first one shows the variance explained by the principal components (Figure 7.4). The window shows a pareto chart with the principal components that explain more than 95% of variance. The application selects the number of PCs that explains more than 90% of variance (the red line in the chart). In the case shown in Figure 7.4, we can see that with the first three PCs the explained variance is 95.6%.

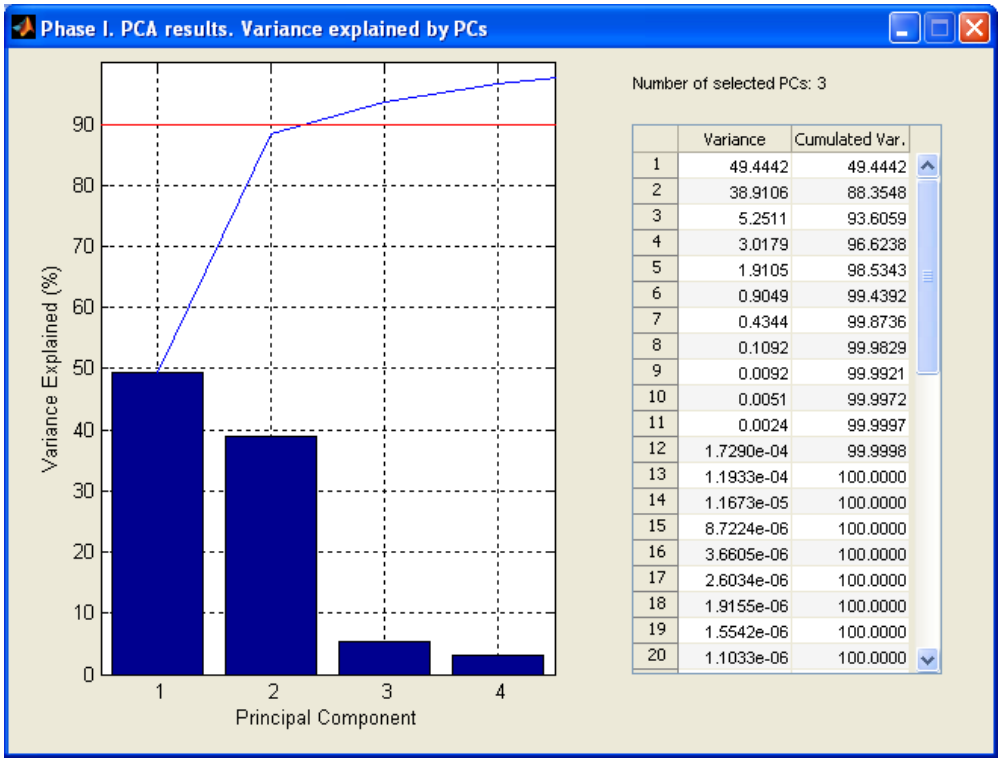


Figure 7.4.- Variance explained by principal components

2. The second window shows the scatter plot for the two first components, with a confidence region at significance level α = 0.05 (red continuous line) and another for α = 0.01 (red dash line), both computed as shown in 6.2.1.1 (Figure 7.5)

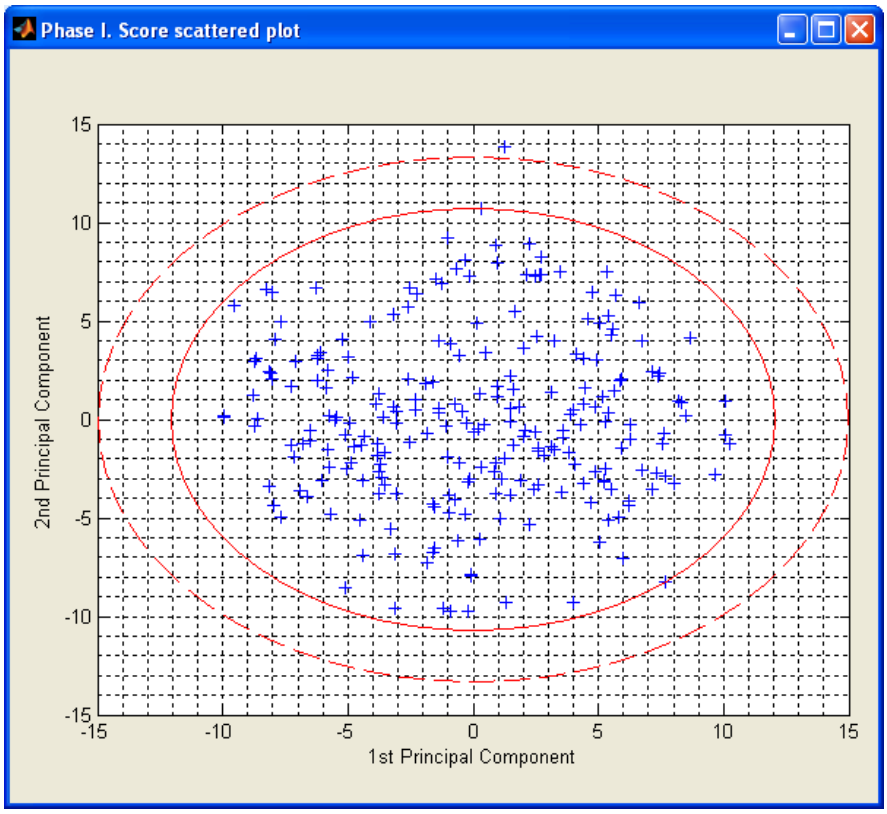


Figure 7.5.- Scatter plot for the two first components.

The user can select any point in this scatter plot to see the variable contributions to this observation in the component selected with the corresponding combo box (Figure 7.6).

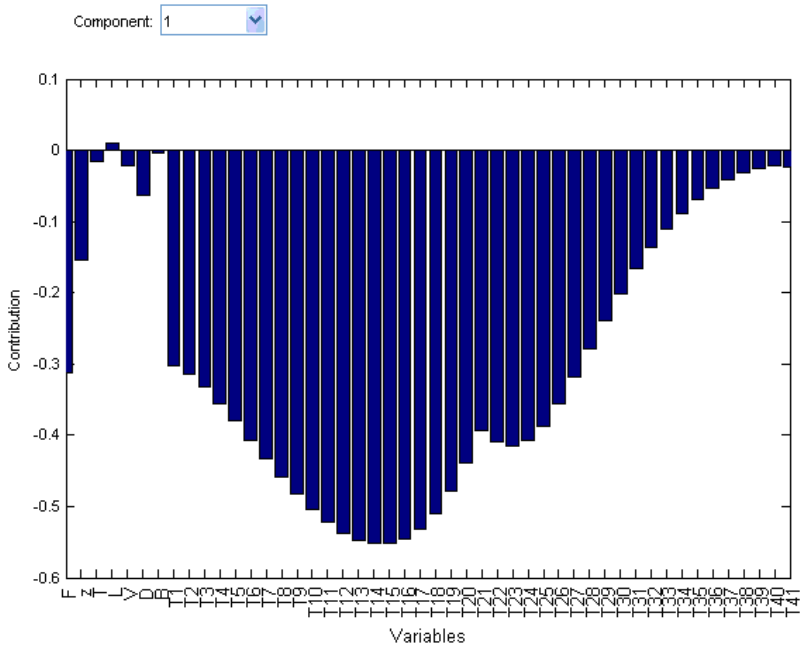


Figure 7.6.- Contribution plot for the selected component.

7.1.3. PCA statistics

The user can plot the SPE chart and the T^2 chart in Phase I (Figure 7.7) 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 significance level α =0.05 (red continuous line) and another for α =0.01 (red dash line).

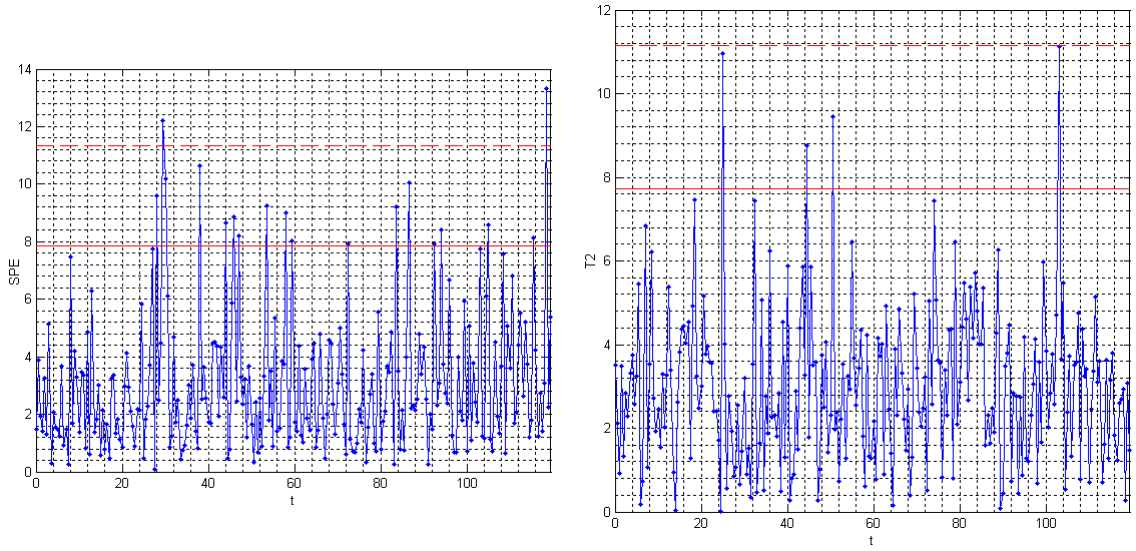


Figure 7.7.- SPE and T^2 charts (Phase 1)

We see some points slightly out of control limits of the SPE and T_A^2 Shewhart chart (at significance level α =0.05) 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. In this case, this allowed number of out of control points is computed as n_{obs} *x 0.05 = 240 x 0.05 = 12*

7.2. Phase II. Model exploitation

7.2.1. Simulation parameters

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 tests in Phase II, through the corresponding "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 Figure 7.9:

1. Simulation time

2. Activate disturbance in z, T 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 Figure 7.8).

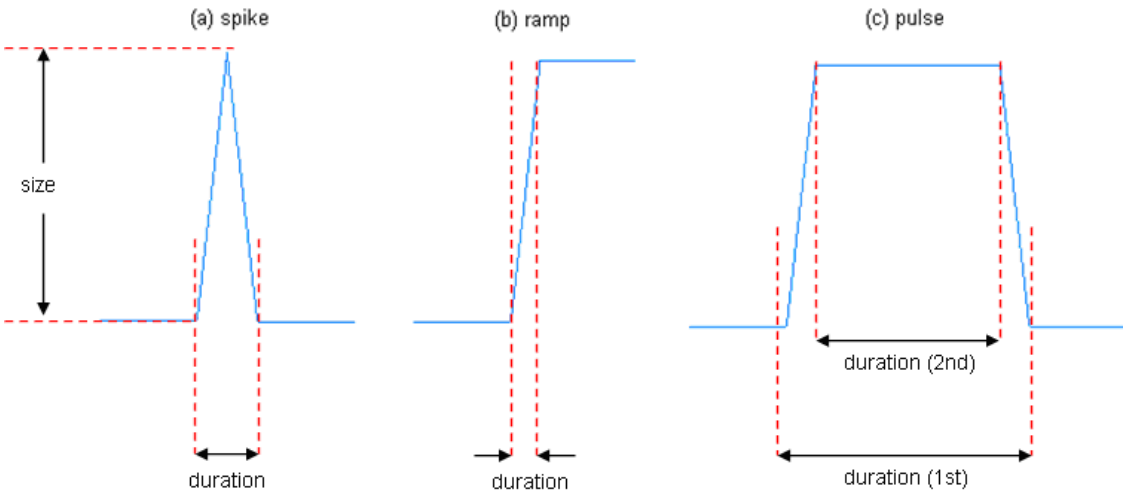


Figure 7.8.- Types of disturbances signals.

3. Activate PI failure (x_B loop or y_D loop). This simulates a failure in the PI control loop. The user just 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.

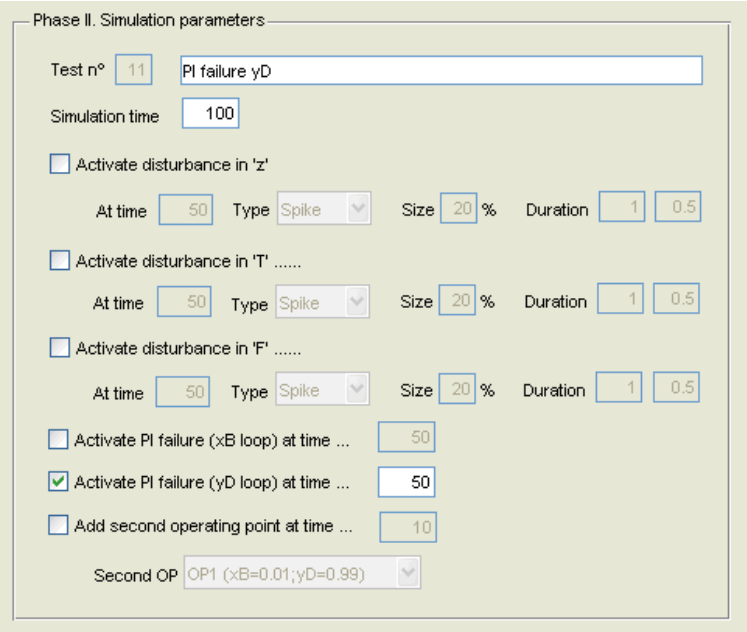


Figure 7.9.- Phase II simulation parameters.

Table 7.2 shows the parameters for the tests done in Phase II for this work. These are several types of signal disturbances and failures in regulatory controls (see chapter 0). In Table 7.2 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 7.2.- Phase II tests parameters.

Test	Simulation time	Disturbance variable	Disturbance time	Signal	Signal duration
1	50	z	10	spike	1
2	50	z	10	ramp	1
3	50	z	10	pulse	1 (0.5)
4	50	T	10	spike	1
5	50	T	10	ramp	1
6	50	T	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	200	z T	10 10	pulse pulse	1 (0.5) 1 (0.5)
15	50	z F	10 10	pulse pulse	1 (0.5) 1 (0.5)
16	50	T F	10 10	pulse pulse	1 (0.5) 1 (0.5)

7.2.2. PCA monitoring

The menu option “Model exploitation” (or the corresponding toolbar button) applies the model built in Phase I to monitor the results of the test of Phase II as explained in 6.2.2. Then, the statistics SPE and Hotelling’s T^2 are computed and shown in charts that are used for process monitoring.

First of all, we run a 100 min simulation with no disturbances (that is, under normal operating conditions) and we check that the model is valid, because no points out of control are shown in any control chart (see Figure 7.10).

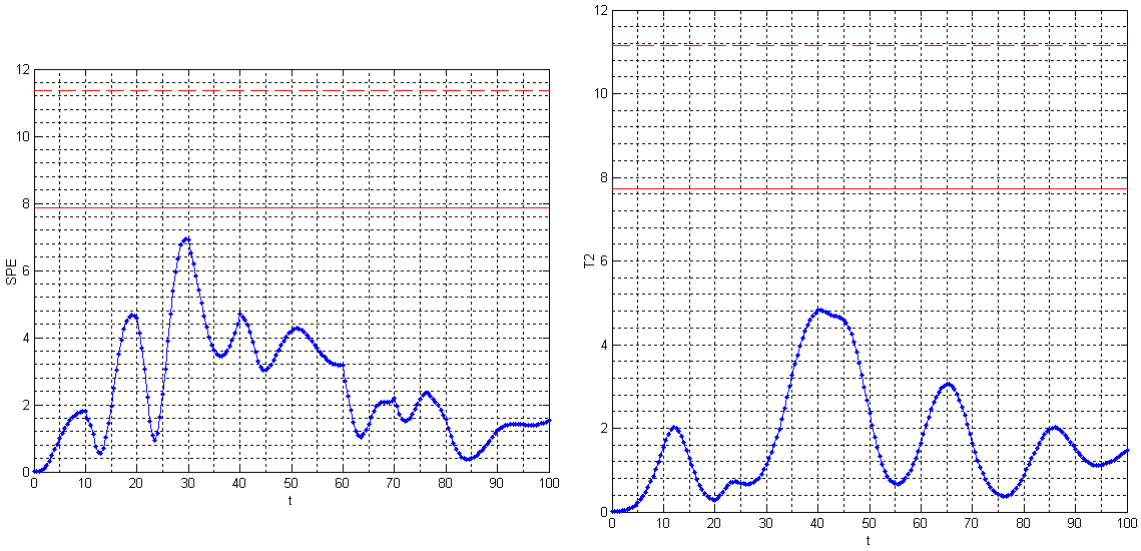


Figure 7.10.- Control charts for process under NOC.

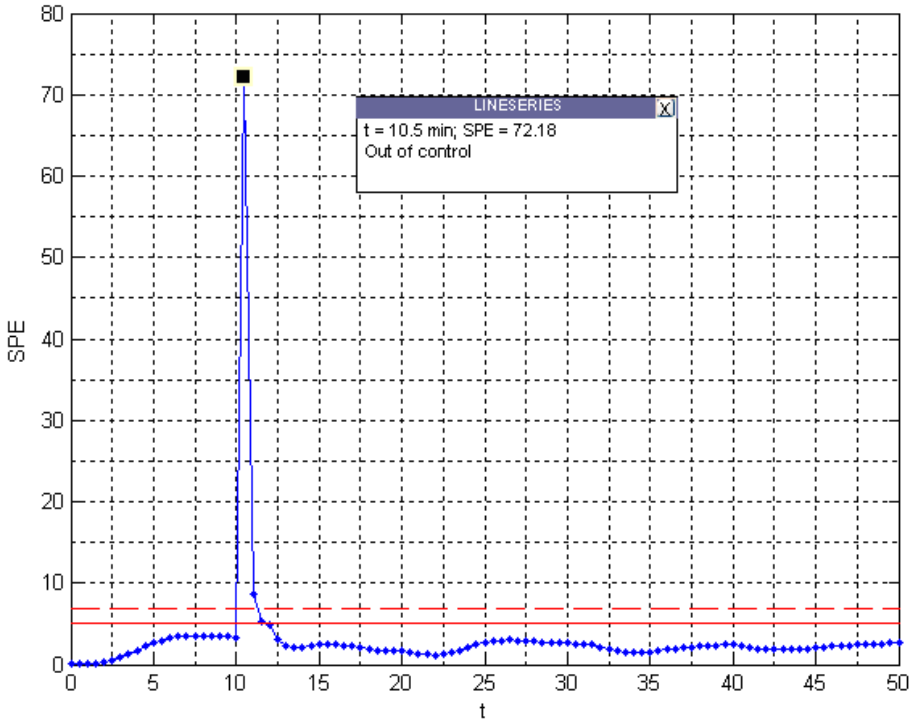


Figure 7.11.- SPE chart for test#1 (z-spike)

Then, we run test#1 in Table 7.2, a 50 min simulation with a spike-type disturbance in composition (z variable) at 10 min.

As stated before (see section 6.2.2.1), first we have to check the SPE control chart, which measures the distance to the model. If there is some point out of control, then we will study its causes with the contribution plot. If there are not, then we must check the T^2 control chart, which measures if the observations are in the zone defined by normal operating conditions.

For example, for test #1 (Figure 7.11) there is an out-of-control point in SPE control chart. The red line is the upper control limit at significance level α = 0.05 and the red dashed line corresponds to the upper control limit at significance level α = 0.01 (see section 6.2.1.3). When the user selects a point in the SPE chart then a contribution plot of the variables for that observation is shown (Figure 7.12, see 6.2.2.2 for computations), which is useful to assess the cause of the failure. If we select the point that is out of control then we can see that the variable that contributes greatly to this out of control observation is variable *z* (feed composition) which is perfectly coherent with the simulated disturbance, that is, the spike signal disturbance at 10 min of simulation.

We have extended the study of the monitoring of a step change in feed parameters (F, z and T) with several types of signals (see tests number 1 to number 9). All of these tests can be monitored with the SPE chart and its corresponding failure analysis can be done with the contribution plots.

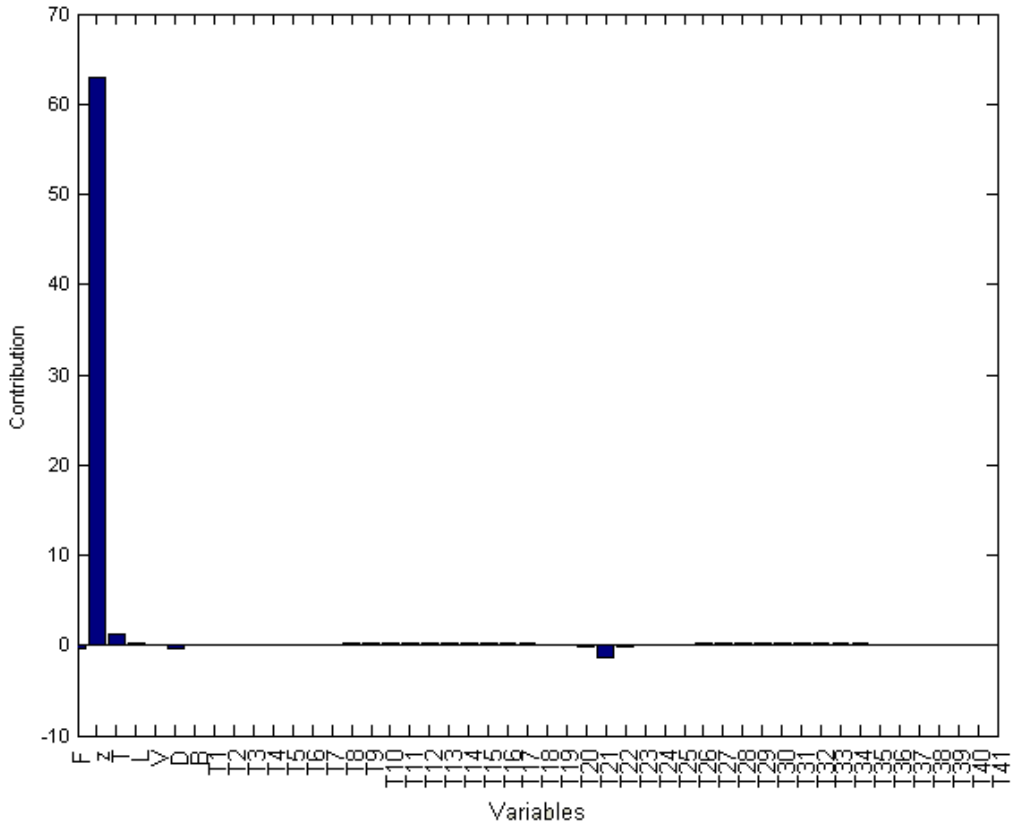


Figure 7.12.- Variables contribution plot (SPE chart)

When the signal is a ramp (see Figure 7.8 for signal types), the disturbance remains at a high level and this is reflected in SPE chart (see Figure 7.13).

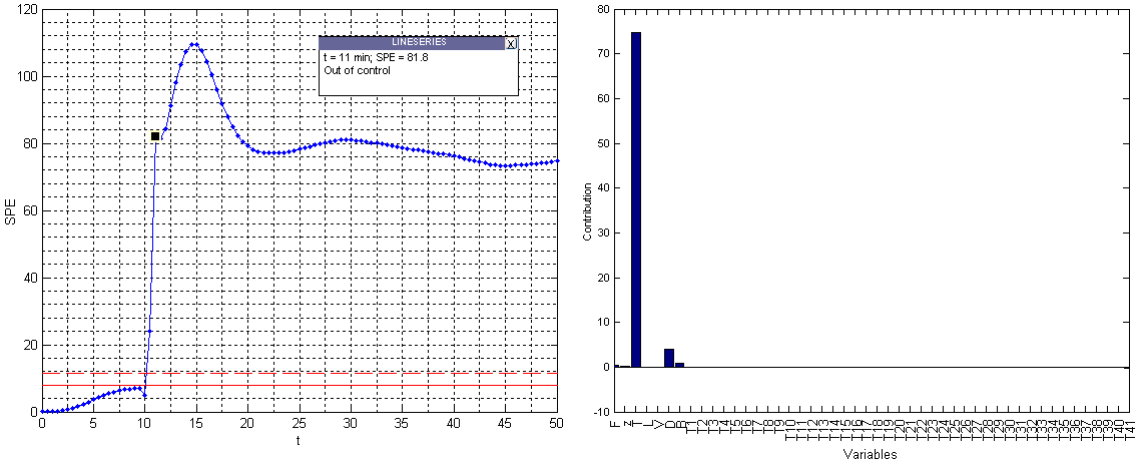


Figure 7.13.- SPE chart for test#5 (T-ramp)

Pulse signals are also detected in SPE chart. Figure 7.14 shows results of test #9, a 50 min simulation with a pulse signal at 10 min for feed flow (F variable).

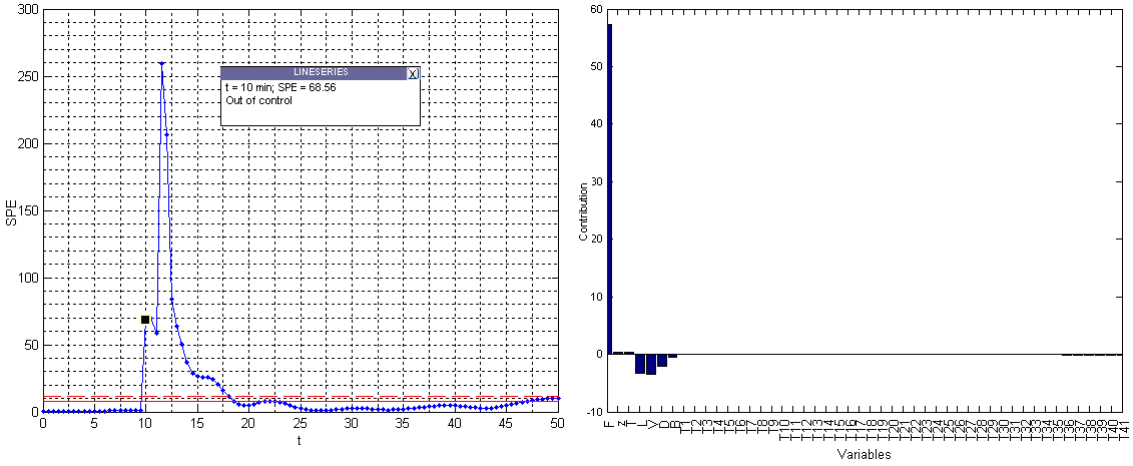


Figure 7.14.- SPE chart for test#9 (F-pulse)

We can see that, initially, the cause of the failure is the high values of feed flow itself but other variables are affected greatly after this disturbance. The contribution plot in Figure 7.15 shows how the bottoms flow B is outside its normal operating values due to the disturbance in feed flow, until the process reaches the steady state after 7 min after the disturbance, approximately.

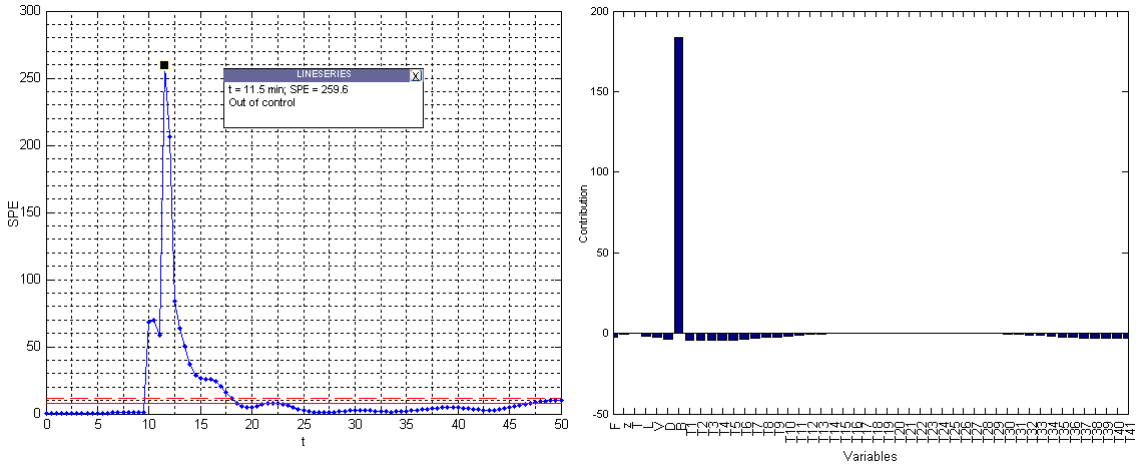


Figure 7.15.- SPE chart for test#9. Effect in flow B.

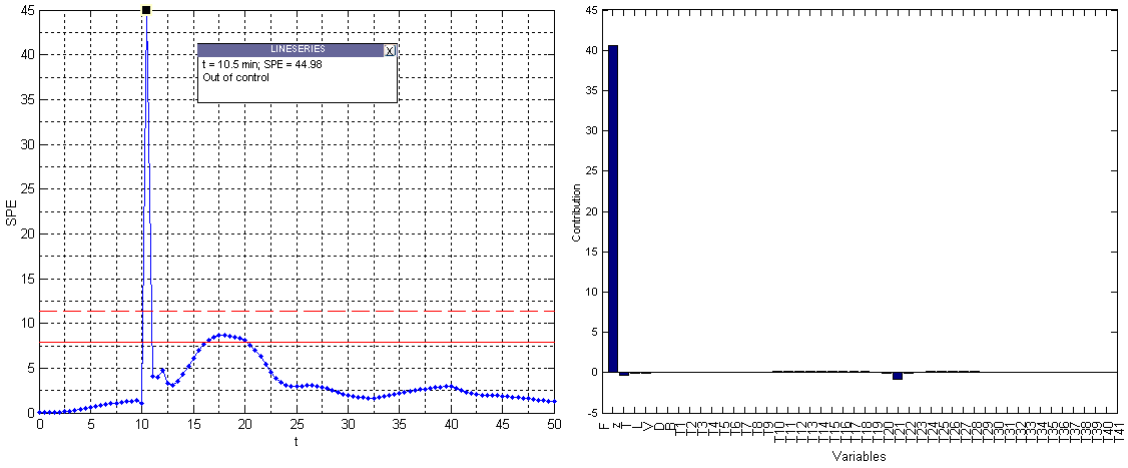


Figure 7.16.- SPE chart for test#1 with a 15% disturbance size

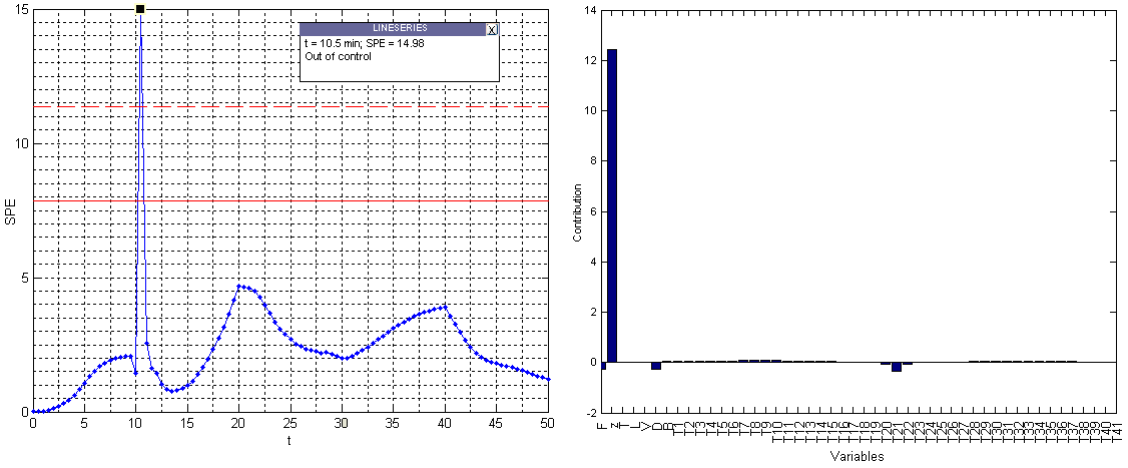


Figure 7.17.- SPE chart for test#1 with a 10% disturbance size

We have repeated test#1 with a disturbance signal of 15% (Figure 7.16) and another one of 10% (Figure 7.17). If we compare them with the disturbance of 20% we can see how SPE value for the point out of control diminishes, but it is still detectable for a 10% disturbance signal.

Another types of failure simulated in this work are related to regulatory controls such the PI for composition control in x_B and y_D (see section 5.3). These types of simulation limit the flow in V (or L) to the mean value under normal conditions. We can see that for test#10 (Figure 7.18), the flow V is limited to a value smaller than the one needed to keep x_B under control. We also see that, due to column interactions, this affects not only the directly related variables (bottoms flow B and x_B composition), but distillate variables D and y_D. In fact, the corresponding SPE chart (Figure 7.19) shows a high contribution of B variable to the selected point out of control, but also a high contribution of D and T1. This variable T1 is the temperature of stage 1 in the distillation column that is directly related to x_B.

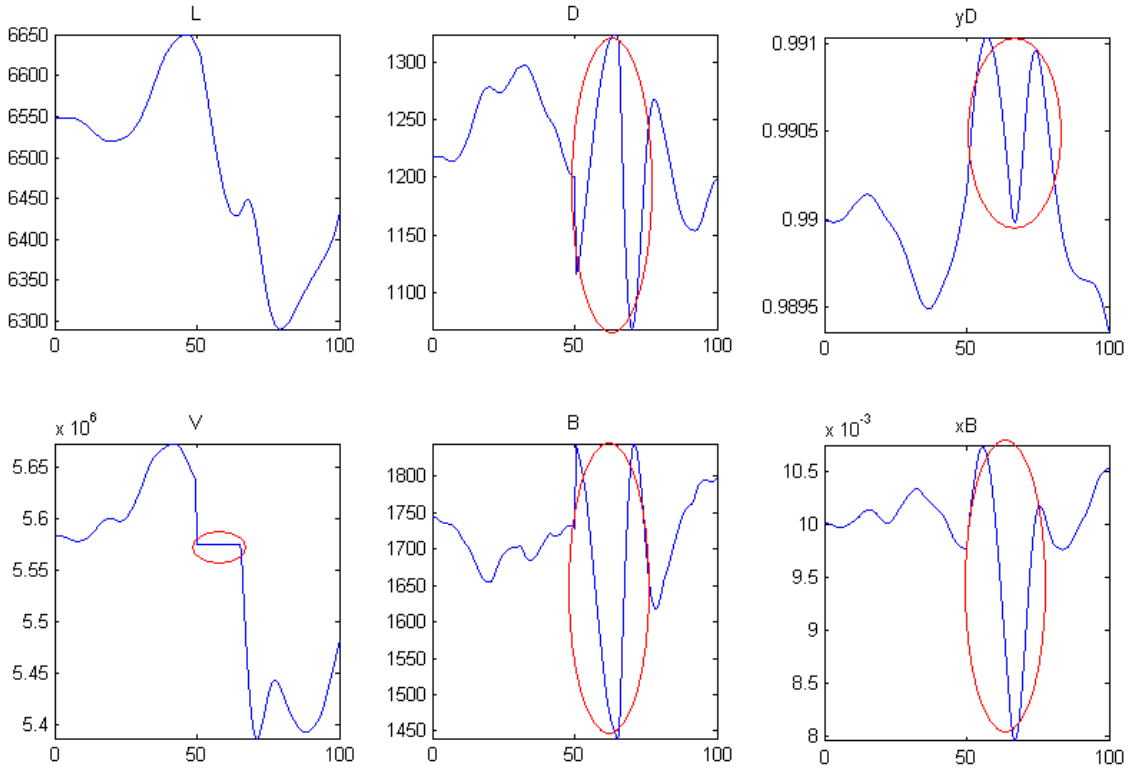


Figure 7.18.- Several variables plots for test#10 (PI failure x_B)

A similar failure is simulated for the y_D control loop (Figure 7.20). In this case the limited flow is L, which affects flows and compositions both in bottoms and distillate, as we can see in the corresponding SPE chart (Figure 7.21).

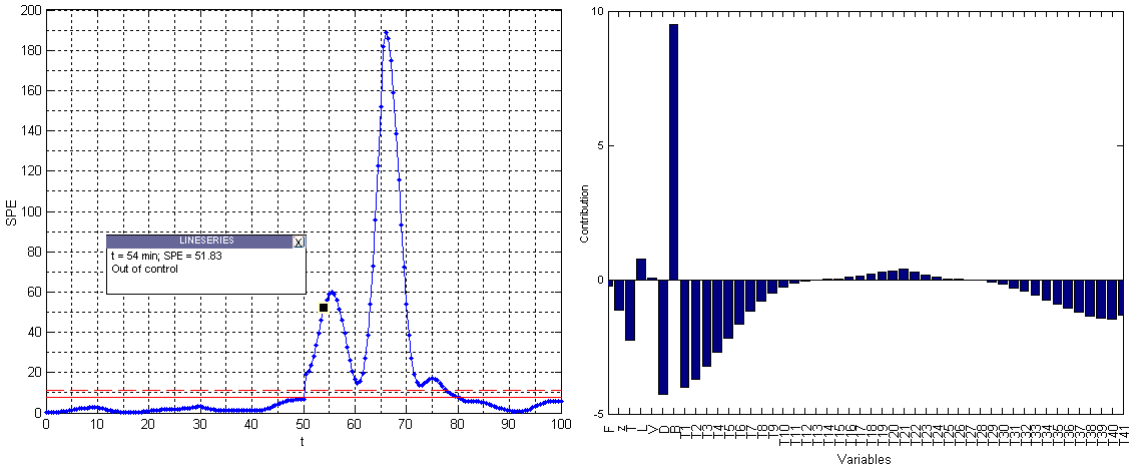


Figure 7.19.- SPE chart for test#10 (PI failure x_B)

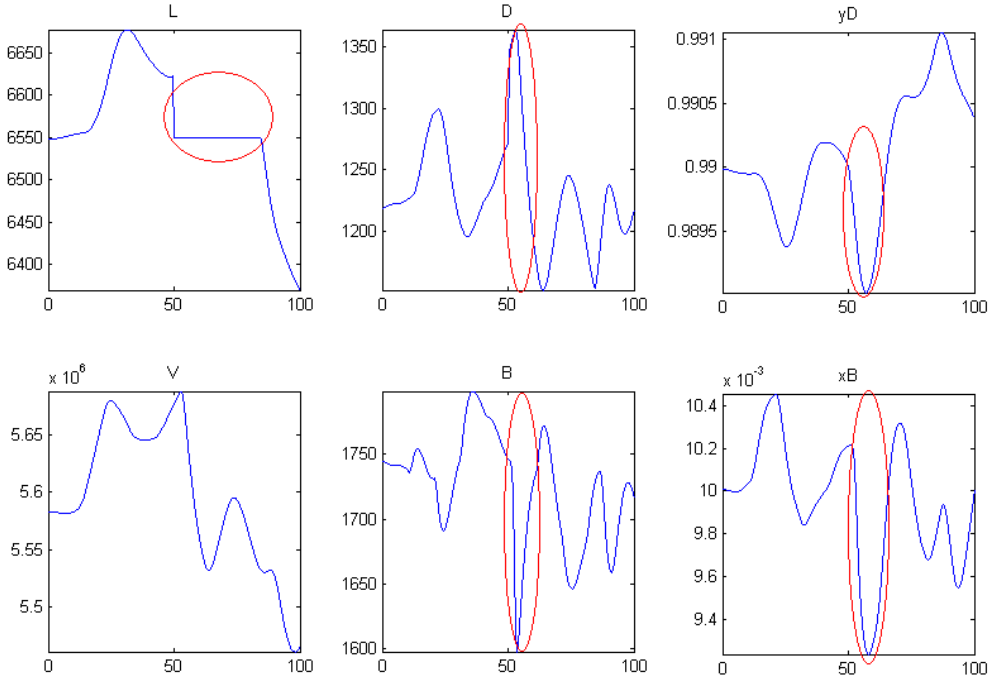


Figure 7.20.- Several variables plots for test#11 (PI failure y_D)

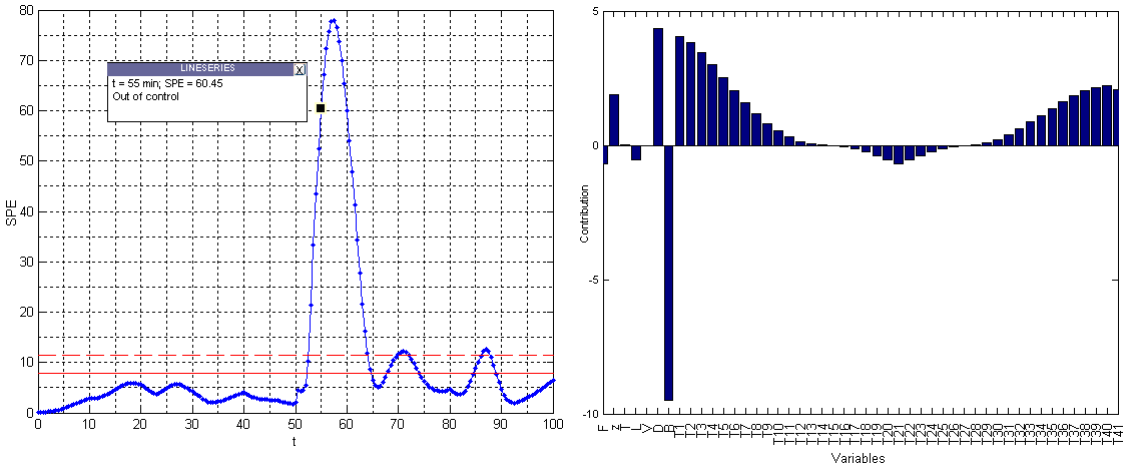


Figure 7.21.- SPE chart for test#11 (PI failure y_D)

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).

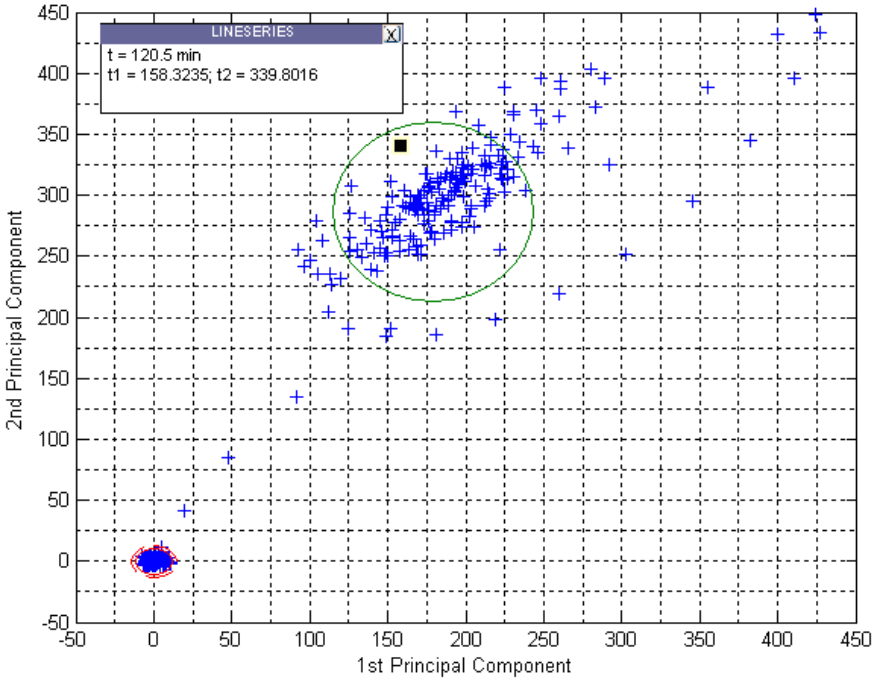


Figure 7.22.- Scattered plot for the first two components for test#12

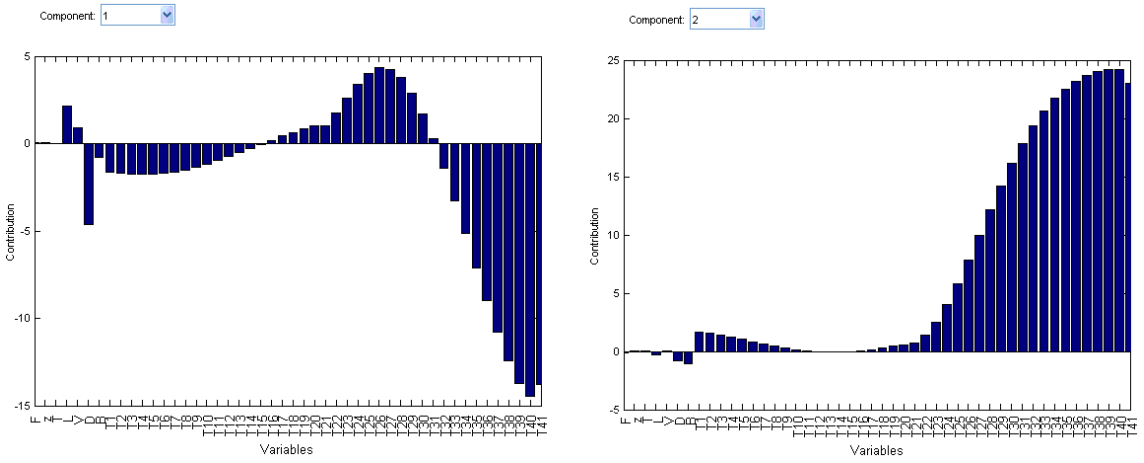


Figure 7.23.- Contribution plots for the first two components (test#12)

Figure 7.22 shows the scattered plot for the first two components in phase II for test#12. We can see 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. I we have a look at the contribution plot for a point of the second cluster (Figure 7.23), we can see 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.

Results for test#13 are quite similar, with two clusters in the scattered plot (Figure 7.24). The first one for the operating point 1 (x_B = 0.01, y_D = 0.99), the one explained by the model, centered at (0,0) coordinates. The other cluster represents operating point 3 (x_B = 0.05, y_D = 0.99) and is far away from our model. In this case, the contribution plot for component 1 and 2 shows a great contribution of the temperatures at the first stages of the distillation column (T1 and following) that are directly related to x_B composition which is 0.05 instead of 0.01 that is the "model composition".

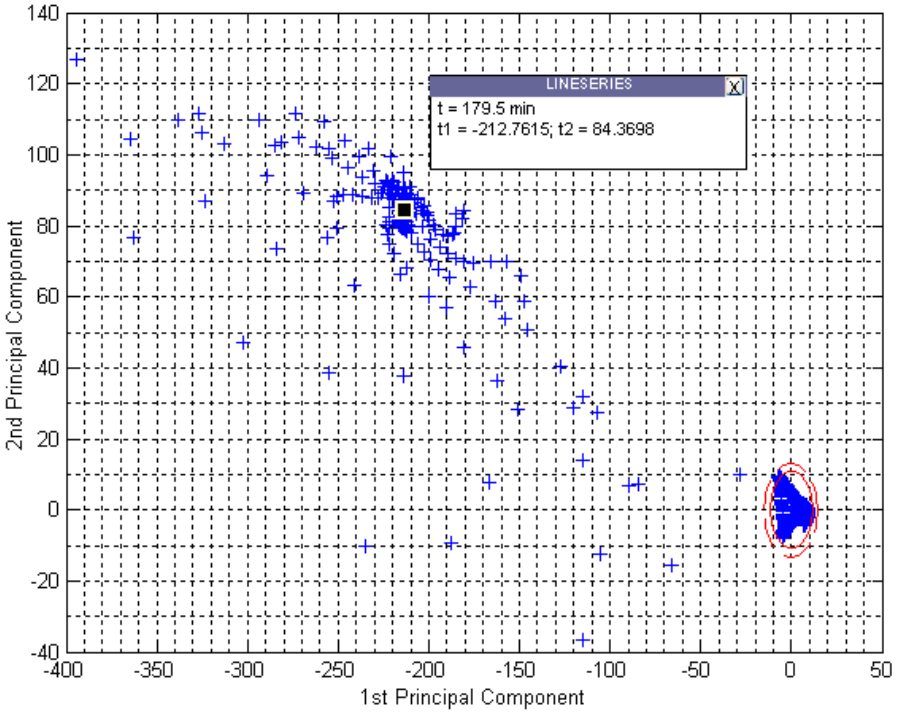


Figure 7.24.- Scattered plot for the first two components for test#13

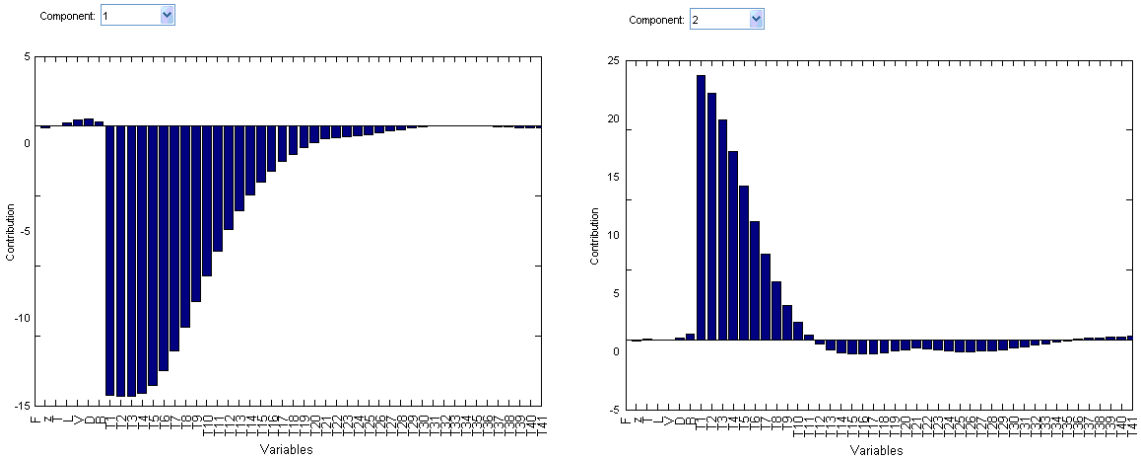


Figure 7.25.- Contribution plots for the first two components (test#13)

In both tests (#12 and #13), both control charts SPE and T^2 can detect the abnormal situation (Figure 7.26). The SPE is a measure of the distance to the model that we have seen that is great for the second cluster of observations. On the other hand the correlation structure of the model is broke down during the transition from one operating point to another and this is reflected in the T^2 control chart. The model obtained during Phase I is not valid to explain the process variability for other operating points. We can see the great values for both statistics due the great difference between both situations.

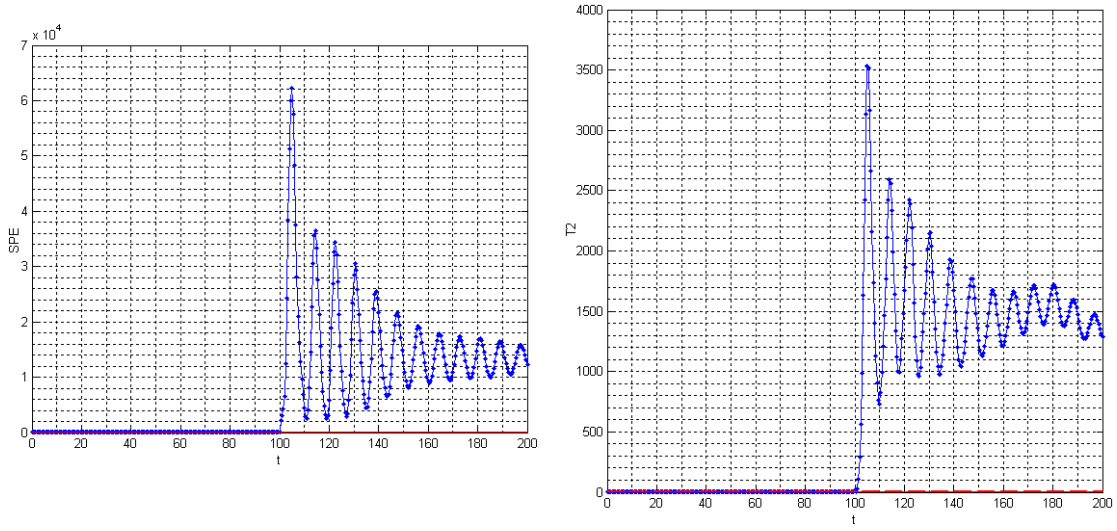


Figure 7.26.- SPE and T2 control charts for test#12

The last group of simulations deals with the capability of PCA monitoring to explain the causes of failures to more complicated situations that the ones presented in test#1 to 9. Here, we will make some combinations of feed disturbances simultaneously (see tests#14 to 16 in Table 7.2). We can see how PCA is able to detect not only the abnormal situation but its root causes (Figure 7.27, Figure 7.28 and Figure 7.29). When F disturbance is present, there is a collateral effect on variable B (pointed out by a red circle in Figure 7.28 and Figure 7.29). This situation is the same as the one shown in Figure 7.15.

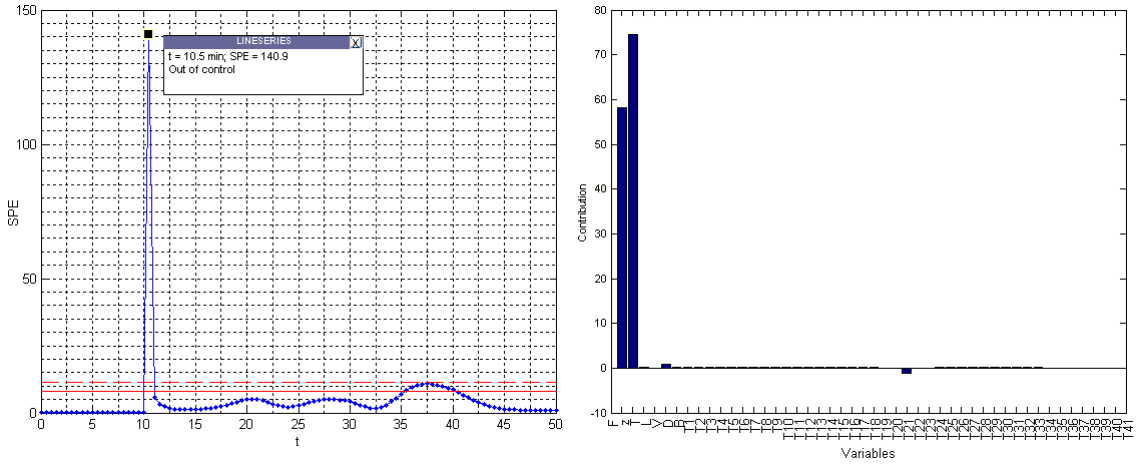


Figure 7.27.- SPE chart and contribution plot for #14 (z&T pulse)

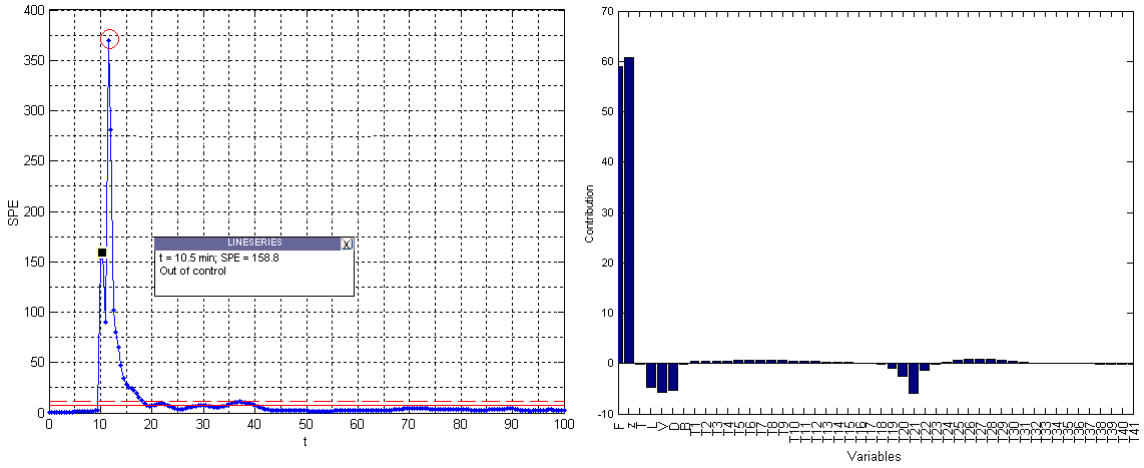


Figure 7.28.- SPE chart and contribution plot for #15 (F&z pulse)

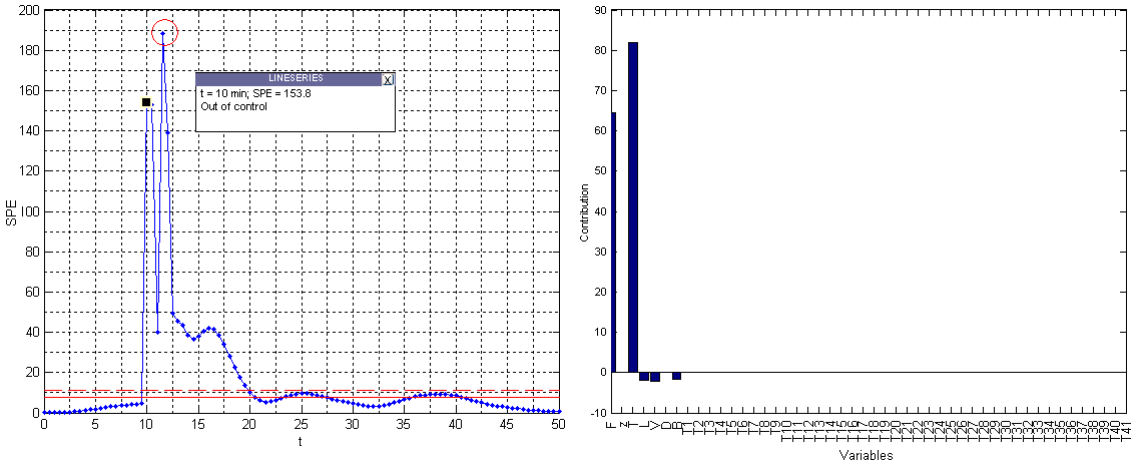


Figure 7.29.- SPE chart and contribution plot for #15 (F&T pulse)

8. Conclusions

The conclusions derived from this work are the following:

1. The original model developed by Skogestad [1] has been modified to obtain a more realistic one. Some model variables are non conventional for industrial measurements so they have been converted to another more common in industrial environments.

2. The model has been studied using different statistical techniques to understand the interaction between the disturbances, manipulated and controlled variables.

   We have demonstrated that flow rate and composition of the feed affects all four manipulated variables (flows *L*, *V*, *D* and B*).* In the case of flows *D* and *B* (the ones used for level control), the interaction effect between flow rate and composition of the feed is slightly significant.

   If feed flow rate increases, all flows (*L*, *V*, *D* and *B*) increase.

   The distillates composition y_D is controlled by manipulating flow *L* and the bottoms composition x_B is controlled with flow *V*. To keep set points, when feed composition increases the PI control systems compensates this by increasing both flows *V* and *L*.

   On the other hand, an increase in feed composition means that the feed enriches in the more volatile compound, so there is more product in the distillate and the level in the distillate drum tends to increase. To keep control level, the proportional control increases the distillate flow *D*

3. Two different optimization methods had been implemented for the tuning of the two PI control loops of the model.

   First of all, a vast study has been done to define the best experiment conditions. It has been studied the simulation time, several response variables (ISE, ITA, IAE) and the step changes in set points. It has also been studied the interaction between both PI control loops.

   The first optimization method implemented is the steepest descent method. This is a single response method that has yield acceptable results. The main drawbacks of this method is the high number of tests needed to find a solution and the fact that it does not have into account the interaction between both control loops x_B and y_D. Due to this interaction, a multiple response method is more convenient.

   The second method is based on mathematical optimization and is able to optimize one of the control loops without penalizing the other, unlike the steepest descent method. A study of the percentage of variation to compute the factor levels is done. The higher this percentage of variation the smaller the R^2 of the fitted model, although it is still acceptable. Unlike the steepest descent method, now we do not need to start with a linear model and we fit directly a quadratic model. That lets us to start with high

percentage of variations of factor levels, that is, to explore a greater zone and to obtain results with fewer iterations.

The obtained results are valid for the experiment used for the optimization, but they may not be acceptable for other situations. In particular, the results for the y_D control loop are very under damped when tested in the set point change in x_B. In that case, we could have improved the results by adding another constraint to the optimization model. This constraint could be based on some kind of function that measures the damping ratio. That way, we could limit this effect.

4. The model has been fully parameterized to ease the creation of 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. A MATLAB interface has been implemented to ease the interaction with all these parameters and to export and manage the simulation results. It is possible to simulate several types of disturbances and failures in the system that can be analyzed afterwards.

5. The software implemented in this work is able to obtain a PCA model for data simulated under normal operating conditions (phase I). Then, a second simulation with some upset conditions or failures is run (phase II) and the PCA model of phase I is used to detect such abnormal situations and check for their root causes.

   A methodology for obtaining an optimized model size is developed. This let us to get a minimum model size (that is, minimum computing requirements) without losing monitoring capabilities.

   When checking control charts, first we check the SPE control chart, which measures the distance to the model. If there is some point out of control, then we will study its causes with the contribution plot. If there are not, then we must check the T^2 control chart, which measures if the observations are in the zone defined by normal operating conditions.

   A first initial study is done for the monitoring of a step change in feed parameters (F, z and T) with several types of signals (spike, pulse and ramp). All of these tests can be monitored with the SPE chart and its corresponding failure analysis can be done with the contribution plots. The limit of detection is also studied with a series of test in which we have varied the signal size. If we compare them with the disturbance of 20% we can see how SPE value for the point out of control diminishes, but it is still detectable for a 10% disturbance signal.

   Another types of failure simulated in this work are related to regulatory controls such the PI for composition control in x_B and y_D. Again, SPE chart is useful to detect and understand these types of failures.

In the case of transitions between operating points, the scattered plot for the first two components in Phase II shows us 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 let us understand that the range of temperatures in the column differ greatly from one operating point to another, so the model developed in Phase I is no longer valid. In both tests, both control charts SPE and T2 can detect the abnormal situation. The SPE is a measure of the distance to the model that we have seen that is great for the second cluster of observations. On the other hand the correlation structure of the model is broke down during the transition from one operating point to another and this is reflected in the T2 control chart.

The last group of simulations deals with the capability of PCA monitoring to explain the causes of failures to more complicated situations. Here, we will make some combinations of feed disturbances simultaneously and we can see how PCA is able to detect not only the abnormal situation but its root causes.

9. Future work

This benchmark could be the basis for several future studies such as the following:

9.1.1. Model complexity

The complexity of the model could be increased by the addition of other models connected to the distillation column. Distillation processes are often used to increase products purity that could not be achieved by reaction. So a typical plant configuration could be a reactor connected to a distillation column:

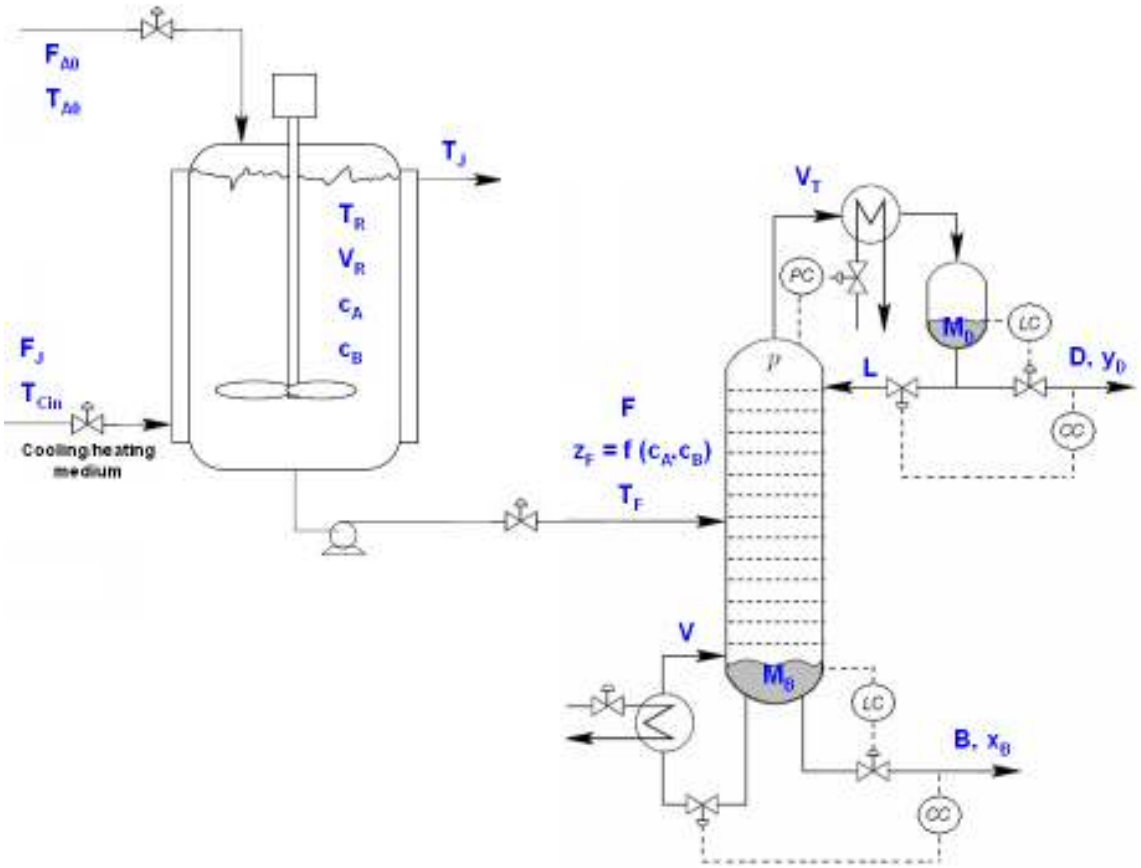


Figure 9.1.- Reactor-column configuration

The reaction that takes place is first order: $A \rightarrow B$

The effluent of the reactor is a mixture of A and B with flow rate *F* at temperature T_F that will be the feed of the distillation column. Luyben's book *Control of CSTR systems* is a great source for reactor models [35]. These models are written in MATLAB code, so they have to be implemented into Simulink functions to be used with the distillation column model.

9.1.2. PI tuning

PID controllers are the most common controllers in industry, in fact, 95% of control loops use PID and the majority is PI. Accordingly, there are many tuning techniques, and most are based on:

1. Empirical methods, such as Ziegler-Nichols methods.
2. Analytical methods, for instance, the root locus based techniques.

Recently, some heuristic optimization techniques have been used for PI tuning purposes [38,39]. In all of these cases, PID tunings are obtained for an operation point where the model can be considered linear. This implies there is sub-optimal tuning when a process operates outside the validity zone of the model. This situation is common when the reference is not a set point but a trajectory (robot control, heating trajectories in furnaces, etc.). An alternative method to solve this problem is to obtain a model for different operational zones, tune a PID controller for each, and establish a mechanism for changing from one controller to another depending on the operation zone (gain planning). Another alternative is tuning a PID controller by taking into account all non-linearities and additional process characteristics. At this point appears the idea of using Genetic Algorithms [40] (a global optimization technique) to obtain a PID tuning that meets all the requirements established in a minimization index by the designer.

9.1.3. Transitions between operating points

Data collected during transitions can be modeled with the same methodology as the data from batch processes [41]. These can be transitions from grade to grade, start-up of a continuous process, and restart of a continuous process that went on hold because of a technical problem. These transitions, which are frequent in plant operations, lead to important loss of production time, large amounts of off-grade materials, and inconsistent reproducibility of product grades. Ideally, optimal transition policies can be obtained using fundamental models and constrained optimization. However, if theoretical models are not available, but a good database on prior transitions and steady states exists, then multivariate *PCA*/*PLS* methods can be used to analyze and improve these transition problems [15].

9.1.4. Controller performance

In a typical continuous process industry facility, there are hundreds to tens of thousands of controllers. These are usually Proportional-Integral (PI) controllers, but can also include nonlinear, adaptive, or multivariable predictive controllers. Any controller is expensive to implement and maintain, requiring hardware, software, and engineering services to generate sustained benefits. Even if these controllers initially perform well, many factors can contribute to their abrupt or gradual performance deterioration, including: sensor/actuator failure, equipment fouling, feedstock variability, product changes, and seasonal influences. As many as 60% of all

industrial controllers have some kind of performance problem. Often these problems fail to attract the attention of personnel who could investigate and address the underlying causes for the performance deterioration [36].

There are many ways to assess the quality of process controllers, but in general they explicitly or implicitly involve a comparison of the current quality of control to some standard. Qin wrote a great review of them with some tutorials and examples [37].

9.1.5. Model Predictive Control

Model predictive control (*MPC*) is an important advanced control technique for difficult multivariable control problems [42,43]. *MPC* does not designate a specific control strategy but a very wide range of control methods that make an explicit use of a model of the process to obtain the control signal by minimizing an objective function. Suppose that we wish to control a multiple-input, multiple-output (*MIMO*) process while satisfying inequality constraints on the input and output variables. If a reasonably accurate dynamic model of the process is available, we can use the model and current measurements to predict future values of the outputs. Then the appropriate changes in the input variables can be calculated based on both predictions and measurements. In essence, the changes in the individual input variables are coordinated after considering the input-output relationships represented by the process model.

MPC can work at different plant levels and can manage from physical to economical variables inside its optimization algorithms. A simple MPC can be implemented to the model shown in this work, which would substitute the two decentralized PI control loops. The model could be improved by simulating condenser and reboiler equipments that would let incorporate economic issues to the optimization phase.

10. Appendix I - Relative volatility

The separation factor or relative volatility α is a numerical measure of the separability of two components by distillation. It is defined as the concentration ratio of *A* and *B* in one phase to that in the other:

$$\alpha = \frac{y/(1-y)}{x/(1-x)} = \frac{y(1-x)}{x(1-y)} \qquad \textbf{Eq. 10.1}$$

The value of α will ordinarily change as *x* varies from 0 to 1.0. If *y* = *x* (except at *x* = 0 or 1), α =1.0 and no separation is possible. The larger the value of α above unity, the greater the degree of separability.

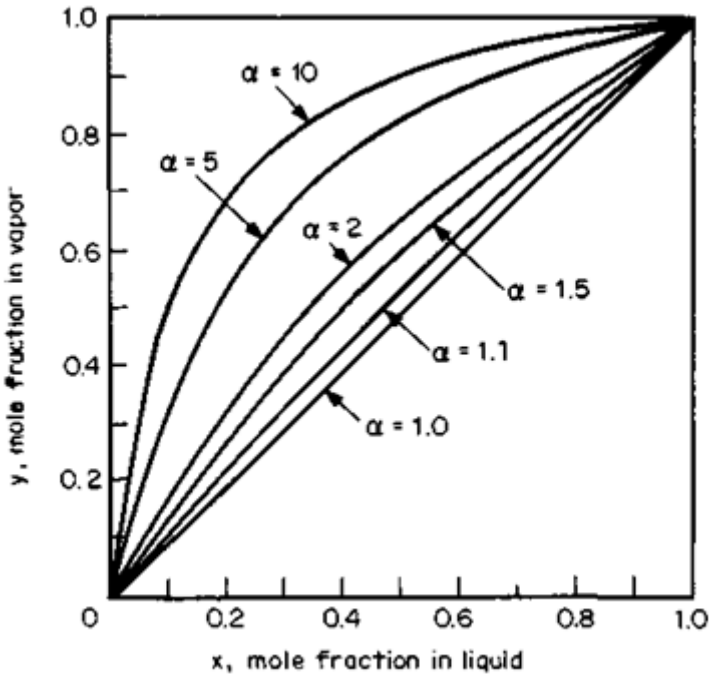


Figure 10.1.- Effect of relative volatility on the concentration in vapor [5]

Figure 10.1 illustrates the effect of relative volatility on the tendency of the more volatile component (*MVC*) to concentrate in the vapor. When volatility is high, the enrichment is large. For instance, when relative volatility is 10, a liquid mixture containing 0.45 mole fraction *MVC* is in equilibrium with vapor containing 0.88 mole fraction of *MVC*; it would take only a few steps to convert the liquid mixture into pure components. Conversely, when relative volatility is very low, say 1.1, a liquid mixture containing 0.45 mole fraction *MVC* is in equilibrium with vapor containing 0.47 mole fraction *MVC*. Under these conditions, it will take a very large number of steps to separate the mixture into the pure components.

11. Appendix II - McCabe-Thiele method

This method calculates the number of theoretical plates of a fractionator needed to achieve a given separation of a binary mixture. Although less rigorous than the method of Ponchon and Savarit, it is nevertheless most useful since it does not require detailed enthalpy data. Except where heat losses or heats of solution are unusually large, the McCabe-Thiele method will be found adequate for most purposes. It is based on the assumptions shown in section 2.2.2 that suppose that, as an approximation, the operating lines on the *xy* diagram can be considered straight for each section of a fractionator between points of addition or withdrawal of streams.

The McCabe-Thiele method integrates the mass balance with the energy balance of the column into something called *operating line* [5]. The column is divided into three operating lines, one for each of the following sections:

11.1.1.1. Rectifying section

For the top section, the total mass balance is:

$$V_i = D + L_{i-1} \qquad \textbf{Eq. 11.1}$$

And for one component:

$$y_i V_i = x_0 D + x_{i-1} L_{i-1} \qquad \textbf{Eq. 11.2}$$

The above equations and the assumption of constant molar flow yield the operating line:

$$y_i = x_0 \frac{1}{1+R} + x_{i-1} \frac{R}{1+R} \qquad \textbf{Eq. 11.3}$$

Where *R=L/D* is the *external reflux ratio* of the column.

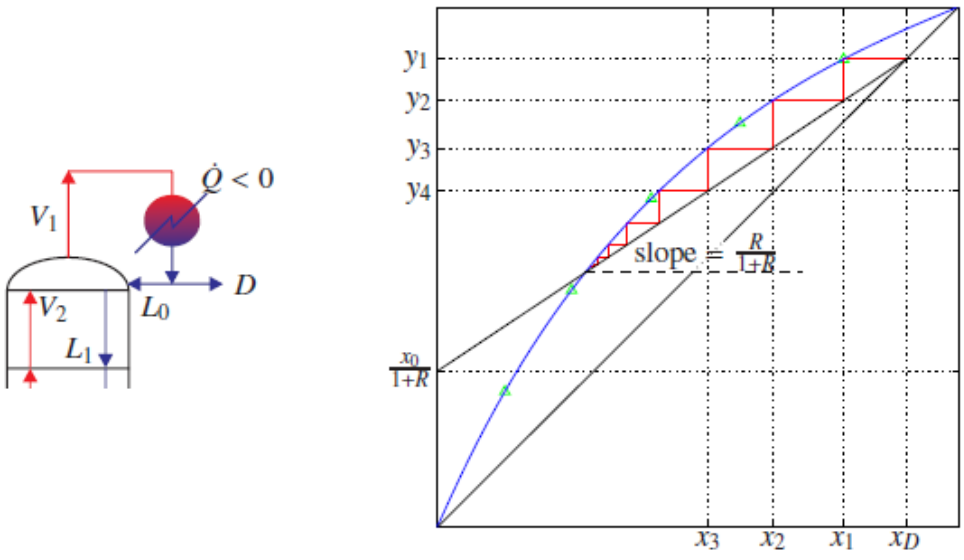


Figure 11.1.- McCabe-Thiele method. Rectifying section.

Figure 11.1 shows the graphical solution of the McCabe-Thiele method in a typical *xy* equilibrium curve for the rectifying section. The staircase (red) shows the interaction between the equilibrium curve (vertical step) and the operating line (horizontal step). Note that the operating line (black) crosses the equilibrium line (blue) at one point in the diagram. The column cannot operate to the left of this crossover point because that would be in the one-phase region of the phase diagram.

11.1.1.2. Stripping section

For the bottom section, the total mass balance is:

$$V_i + B = L_{i-1} \qquad \textbf{Eq. 11.4}$$

And for one component:

$$y_i V_i + x_B B = x_{i-1} L_{i-1} \qquad \textbf{Eq. 11.5}$$

The above equations and the assumption of constant molar flow yield the operating line:

$$y_i = -x_B \frac{1-S}{S} + x_{i-1} \frac{1}{S} \qquad \textbf{Eq. 11.6}$$

Where $S = V/(V+B)$ represents the fraction of the downcoming liquid which is being evaporated.

Figure 11.2 shows the operating line for the stripping section in a typical *xy* equilibrium curve.

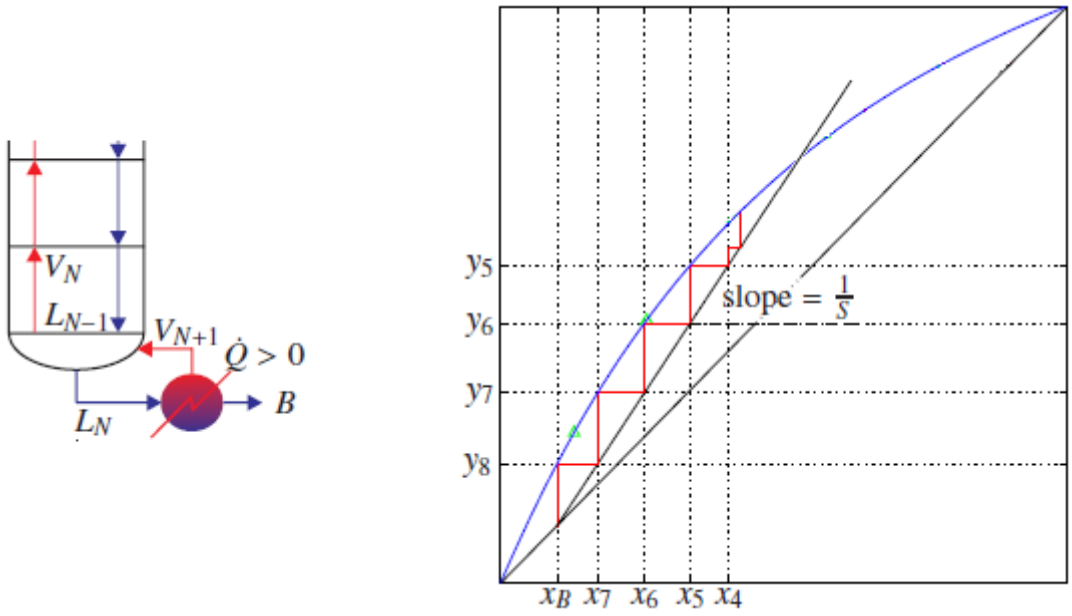


Figure 11.2.- McCabe-Thiele method. Stripping section.

11.1.1.3. Feed tray

For the feed tray section (Figure 11.4), it can be proved that the operating line has the form:

$$y = x\frac{q}{q-1} - x_F\frac{1}{q-1} \qquad \textbf{Eq. 11.7}$$

Where $q = \dfrac{\Delta L}{\Delta L - \Delta V}$ is the *feed quality* (see section 2.2.4).

If we represent all three operating lines into an equilibrium curve, we can calculate graphically the number of theoretical plates needed for the column (Figure 11.4). We can see how the three lines converge at the feed tray.

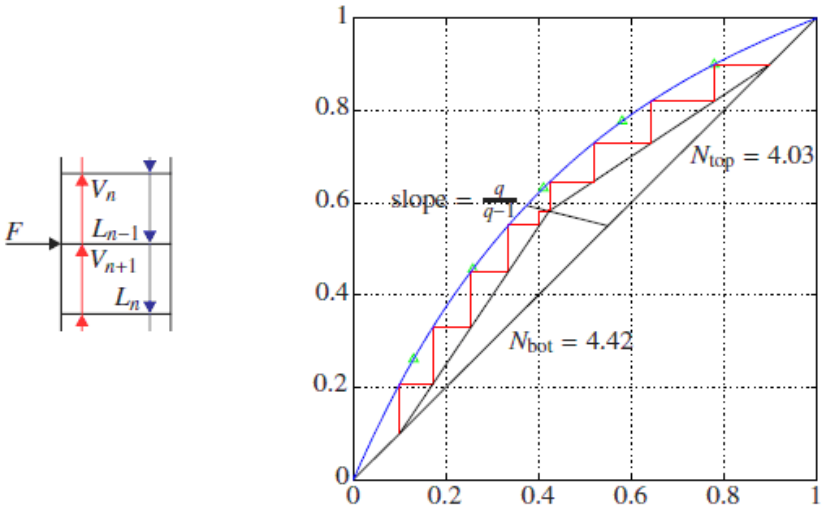


Figure 11.3.- McCabe-Thiele method. Feed tray section.

For the column configuration used in this work (see section 2.2.3) the three operating lines are:

1. Rectifying section

   Replacing $R = \dfrac{L}{D} = \dfrac{2.6889}{0.5} = 5.3778$ and $x_0 = y_D = 0.99$ in Eq. 11.3 we have that the operating line for the rectifying section is:

$$y_i = 0.15523 + 0.84321\, x_{i-1} \qquad \textbf{Eq. 11.8}$$

2. Stripping section

   Replacing $S = \dfrac{V}{V+B} = \dfrac{3.2294}{3.2294+0.5} = 0.86593$ and $x_B = 0.01$ in Eq. 11.6 we have that the operating line for the stripping section is:

$$y_i = -0.0015483 + 1.1548\, x_{i-1} \qquad \textbf{Eq. 11.9}$$

3. Feed tray

   Replacing q = 1.0405 and x_F = 0.5 in Eq. 11.7 we have that the operating line for the feed tray is:

$$y = -12.346 + 25.691\,x \qquad \textbf{Eq. 11.10}$$

Drawing these operating lines over a *xy* diagram for $\alpha = 1.5$ and applying the McCabe-Thiele method (see Appendix II - McCabe-Thiele method), we find that the column has *NT*=41 stages including reboiler and total condenser and the feed is at stage *NF*=21 counted from the bottom (Figure 11.4).

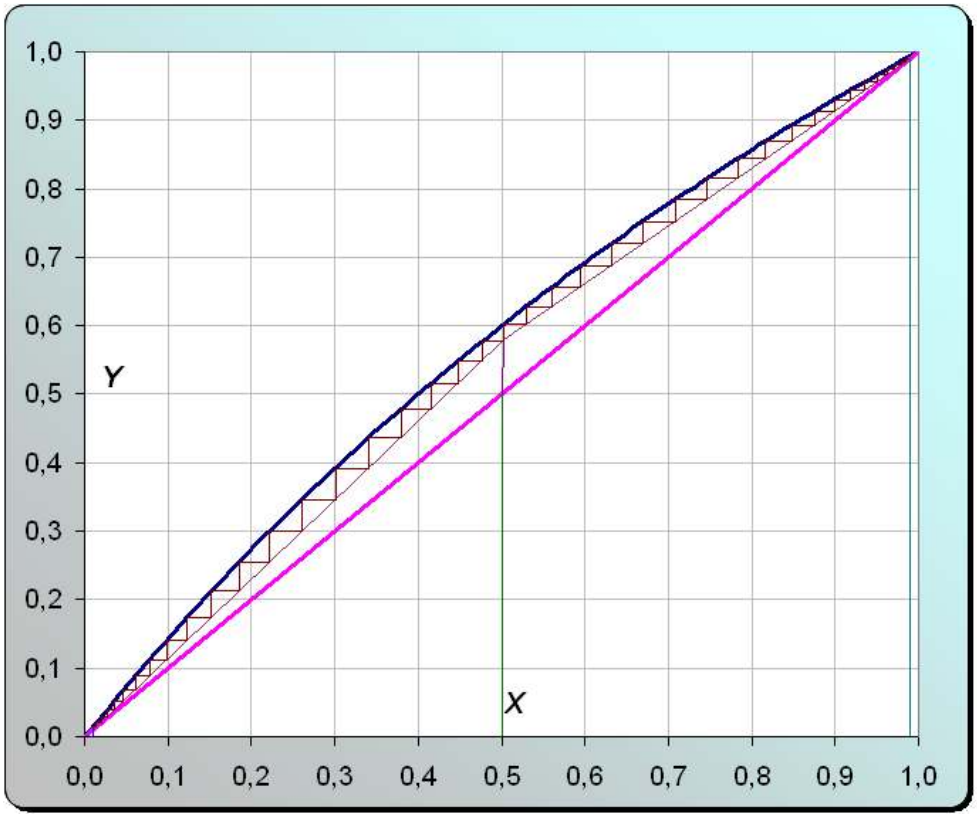


Figure 11.4.- Graphical computation of column stages (McCabe-Thiele method).

12. Appendix III - Feed quality

Consider the section of the column at the tray where the feed is introduced (Figure 12.1). The quantities of the liquid and vapor streams change abruptly at this tray, since the feed may consist of liquid, vapor, or a mixture of both.

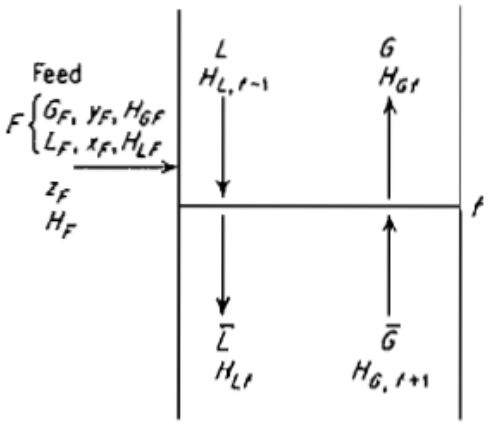


Figure 12.1.- Feed tray [5]

To establish the general relationship, an overall material balance about this section is:

$$F + L + \overline{G} = G + \overline{L} \qquad \textbf{Eq. 12.1}$$

and an enthalpy balance,

$$FH_F + LH_{L_{f-1}} + \overline{G}H_{G_{f+1}} = GH_{G_f} + \overline{L}H_{L_f} \qquad \textbf{Eq. 12.2}$$

The vapors and liquids inside the tower are all saturated, and the molal enthalpies of all saturated vapors at this section are essentially identical since the temperature and composition changes over one tray are small. The same is true of the molal enthalpies of the saturated liquids, so that $H_G = H_0$ and $H_{Lf-1} = H_{Lf}$. Eq. 12.2 then becomes

$$\left(\overline{L} - L\right)H_L = \left(\overline{G} - G\right)H_G + FH_F \qquad \textbf{Eq. 12.3}$$

Combining this with Eq. 12.1 gives

$$\frac{\overline{L} - L}{F} = \frac{H_G - H_F}{H_G - H_L} = \frac{\text{heat to convert to saturated vapor}}{\text{heat of vaporization}} = q \qquad \textbf{Eq. 12.4}$$

The quantity q is thus seen to be the heat required to convert 1 mol of feed from its condition H_F to a saturated vapor, divided by the molal latent heat H_G - H_L. The feed may be introduced under any of a variety of thermal conditions ranging from a liquid well below its bubble point to a superheated vapor, for each of which the value of q will be different. Table 12.1 summarizes values of q factor depending on the thermal conditions of the feed.

The feed of the distillation column simulated in this work is a mixture of methanol and ethanol with z_F molar composition at temperature T_F. Since methanol is more volatile it will be

component *A* and ethanol, component *B*. Table 12.2 shows several thermodynamic properties of both components needed for *q* factor calculations.

Table 12.1.- Values of *q* depending on feed condition.

Feed condition	*q*	Case
Subcooled liquid	>1	(1) $T_F < T_b$
Saturated liquid	1	(2) $T_F = T_b$
Vapor-liquid mixture	0<*q*<1	(3) $T_b < T_F < T_d$
Saturated vapor	0	(4) $T_F = T_d$
Superheated vapor	<0	(5) $T_F > T_d$

Table 12.2.- Thermodynamic properties of methanol and ethanol.

Property	Description	Units	Methanol	Ethanol
M	Molar weight	g/mol	32.04	46.07
C_{pL}	Heat capacity (liquid)	J/(mol K)	79.5	112.4
C_{pV}	Heat capacity (vapor)	J/(mol K)	52.29	78.28
T_n	Normal boiling point	K	337.9	351.4
T_c	Critical temperature	K	513.2	516.2
H_v	Vapor enthalpy at T_n	kJ/mol	35.21	38.56

For sake of simplicity, C_p and H_V will be considered temperature-independent. The mixture is considered ideal and C_p and H_V will be calculated as a weighted mean:

$$C_P = \sum_i x_i C_{Pi} \qquad \textbf{Eq. 12.5}$$

In order to identify the thermal condition of the feed, we need to know the boiling point (T_b) and dew point temperatures (T_d) of the mixture, which will be estimated from the *Txy* diagram. The *Txy* diagram is built from the vapor pressures of the mixture at different compositions. Based on Antoine equation, the vapor pressure p^* (in kPa) for a pure component is given by:

$$\ln p^* = A - \frac{B}{t_{bp} + C} \qquad \textbf{Eq. 12.6}$$

or the equivalent equation,

$$t_{bp} = \frac{B + C\left(\ln p^* - A\right)}{A - \ln p^*} \qquad \textbf{Eq. 12.7}$$

where t_{bp} is the boiling point temperature (ºC) and *A, B* and *C* are the Antoine coefficients. In this case,

Table 12.3.- Antoine coefficients for methanol and ethanol.

Compound	Antoine coefficients		
	A	***B***	**C**
Methanol	16.5785	3638.27	239.500
Ethanol	16.8958	3795.17	230.918

According to Raoult's Law, partial vapor pressures of an ideal mixture of *A* and *B* volatile compounds are:

$$p_A = x_A p_A^* \qquad \textbf{Eq. 12.8}$$

$$p_B = x_B p_B^* = (1 - x_A) p_B^* \qquad \textbf{Eq. 12.9}$$

where p^*_i is the vapor pressure of component *i* at a given temperature.

On the other hand, the column will operate at atmospheric pressure, so

$$p_{total} = p_A + p_B = 760 \text{ mm Hg} \qquad \textbf{Eq. 12.10}$$

So we have a system of four equations (Eq. 12.7 through Eq. 12.10) and six variables: *x,* p_a^*, p_b^*, p_a, p_b and t_{bp} (*T*). The methodology to obtain the bubble point line (*T vs. x*) of the *Txy* diagram is based on fixing a *x* value and solving the nonlinear system to obtain temperature *T*. Repeating these steps for different values of *x* renders the liquid line (*T vs. x*).

If we assume an ideal mixture in vapor phase, partial vapor pressure could be written as:

$$p_A = y_A p_{total} \quad \Rightarrow \quad y_A = p_A / p_{total} \qquad \textbf{Eq. 12.11}$$

For each pair *(T,* x_A*)* of points in liquid line, we calculate p_A:

With *T* (t_{bp}) and Eq. 12.6 we calculate p_A^*

With p_A^*, x_A and Eq. 12.8 we obtain p_A

Then, with p_A, p_{total} (760 mm Hg) and Eq. 12.11 we calculate the corresponding *y* value for the vapor line *(T vs. y)*:

Figure 12.2 shows methanol-ethanol *Txy* diagram obtained with above methodology and a 0,01 step size in mole fraction (*i*.e., 100 points between 0 and 1).

As we have seen, five situations may occur if we take into account feed condition (Table 12.1). Now we are able to identify these cases in a *Txy* diagram, depending on mole fraction (z_F) and temperature of the feed (T_F). Given a feed composition, we can establish the temperature at which first bubble of vapor appears (bubble point temperature, T_b, case 2 in Figure 12.2). That situation corresponds to a saturated liquid. The temperature at which last drop of liquid evaporates (dew point temperature, T_d, case 4 in Figure 12.2) belongs to a saturated vapor. Below bubble point there is only subcooled liquid (case 1 in Figure 12.2), between bubble point

and dew point there is a mixture of liquid and vapor (case 3 in Figure 12.2) and, above dew point, only superheated vapor exists (case 5 in Figure 12.2). Table 12.1 summarizes all these situations.

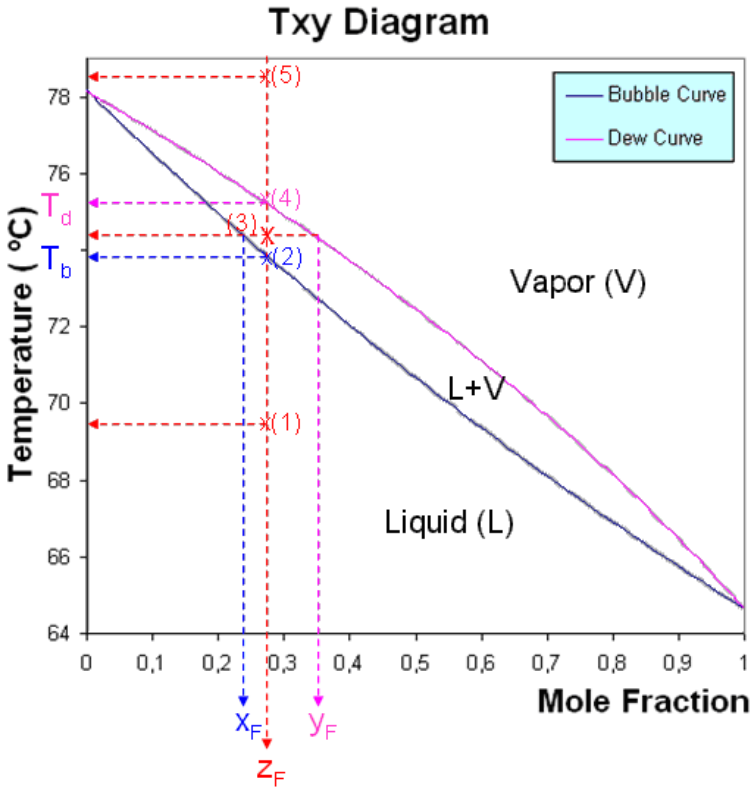


Figure 12.2.- Methanol-Ethanol *Txy* diagram

The simulation will only take into account subcooled liquid feed and *q* factor will be calculated following these steps:

1. Given a feed with composition z_F, from the *Txy* diagram, calculate bubble point temperature (T_b) and dew point temperature (T_d).

   For example, for z_F = 0.5, T_b = 70.67ºC and T_d = 72.45ºC (obtained from the *Txy* diagram in Figure 12.2). Temperature of the feed is not considered here because it is a subcooled liquid: T_F is less than T_b and for any feed temperature below T_b with a determined composition, T_b and T_d will always be the same.

2. If heats of solution are ignored, the enthalpy of the feed at T_b (bubble point of the feed) referred to T_F (the feed temperature) is:

$$H_L = C_{pL,F}\left(T_b - T_F\right) \qquad \textbf{Eq. 12.12}$$

   $C_{pL,F}$ can be calculated as a weighted mean, that is $C_{pL,F} = C_{pL,A}\,z + C_{pL,B}\left(1-z\right)$

For example, for a feed temperature of 55ºC and z_F = 0.5, the enthalpy of the feed will be:

$$H_L = \left[79.5 \cdot 0.5 + 112.4\left(1 - 0.5\right)\right]\left(70.67 - 55\right) = 1503.54\ J/mol$$

3. Calculate the heats of vaporization for component *A* and *B* at temperature T_b. Here we use the Watson equation [4], that estimates the heat of vaporization of a liquid at any temperature from its heat of vaporization at another temperature:

$$\frac{H_{v2}}{H_{v1}} = \left(\frac{T_C - T_2}{T_C - T_1}\right)^{0.38} \qquad \textbf{Eq. 12.13}$$

where,

H_{v1}: Heat of vaporization of the liquid at T_1, in J/mol

H_{v2}: Heat of vaporization of the liquid at T_2, in J/mol

T_1: Temperature, in K

T_2: Temperature, in K

T_c: Critical temperature of the liquid, in K

In Table 12.2 we have the heat of vaporization at normal boiling point, so we can compute the heat of vaporization at $T_2 = T_b$ = 70.67 ºC = 343.8 K:

$$H_{v2,A} = 35.21\left(\frac{513.2 - 343.8}{513.2 - 337.9}\right) = 34.753\ kJ/mol = 34753\ J/mol$$

$$H_{v2,B} = 38.56\left(\frac{516.2 - 343.8}{516.2 - 351.4}\right) = 39.225\ kJ/mol = 39225\ J/mol$$

4. The enthalpy of the saturated vapor at T_d (dew point of the feed) referred to liquid at T_F (the feed temperature) is:

$$H_G = z_F\left[C_{pl,A}\left(T_d - T_F\right) + H_{v,A}\right] + \left(1 - z_F\right)\left[C_{pl,B}\left(T_d - T_F\right) + H_{v,B}\right] \qquad \textbf{Eq. 12.14}$$

In our example:

$$H_G = 0.5\left[79.5\left(72.45 - 55\right) + 34753\right] + \left(1 - 0.5\right)\left[112.4\left(72.45 - 55\right) + 39225\right] = 38663\ J/mol$$

5. For a subcooled liquid there is no vapor phase, so H_F=0

6. Calculate *q* factor with Eq. 12.4.

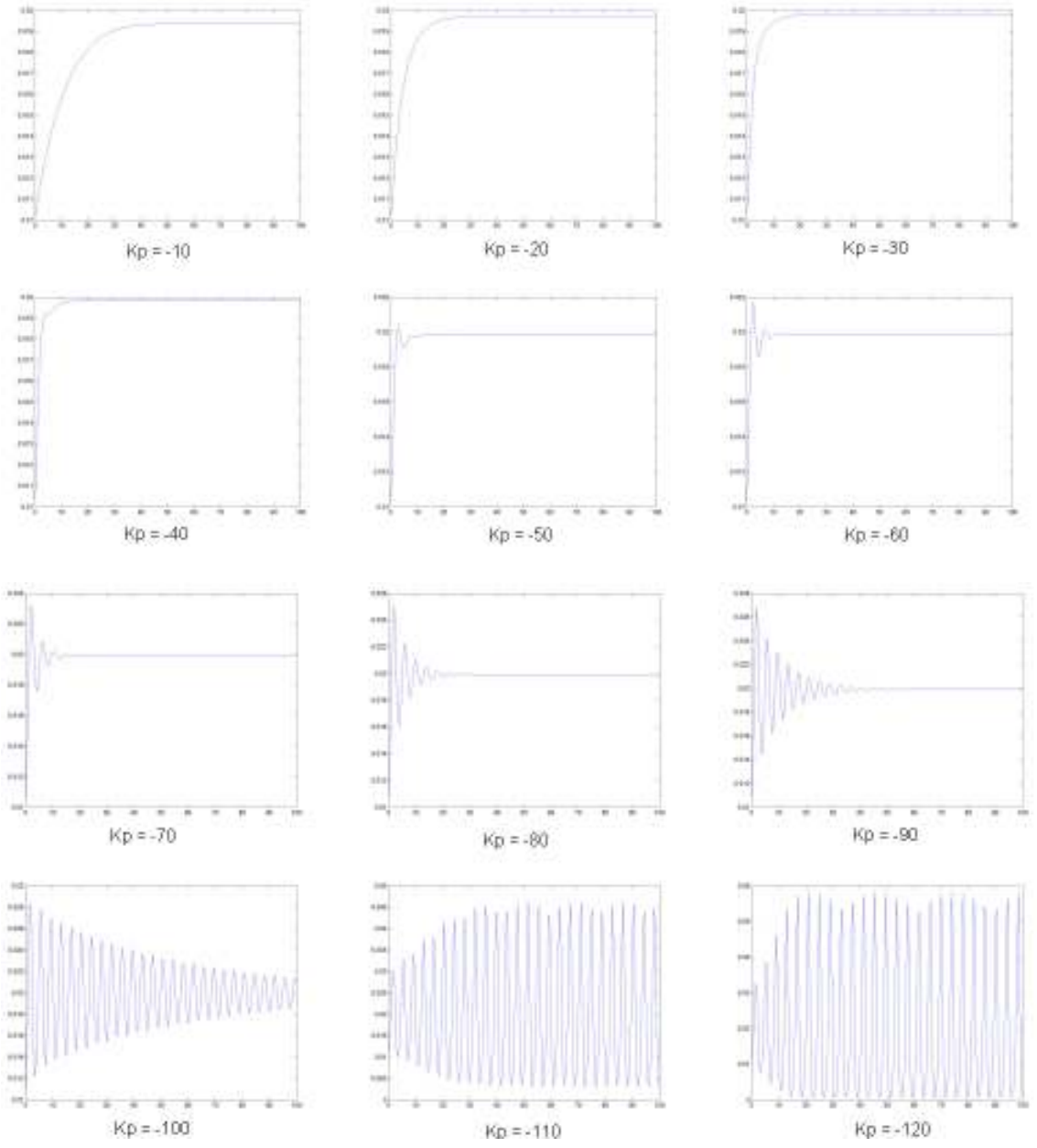
In our example,

$$q = \frac{H_G - H_F}{H_G - H_L} = \frac{38663 - 0}{38663 - 1503.54} = 1.0405$$

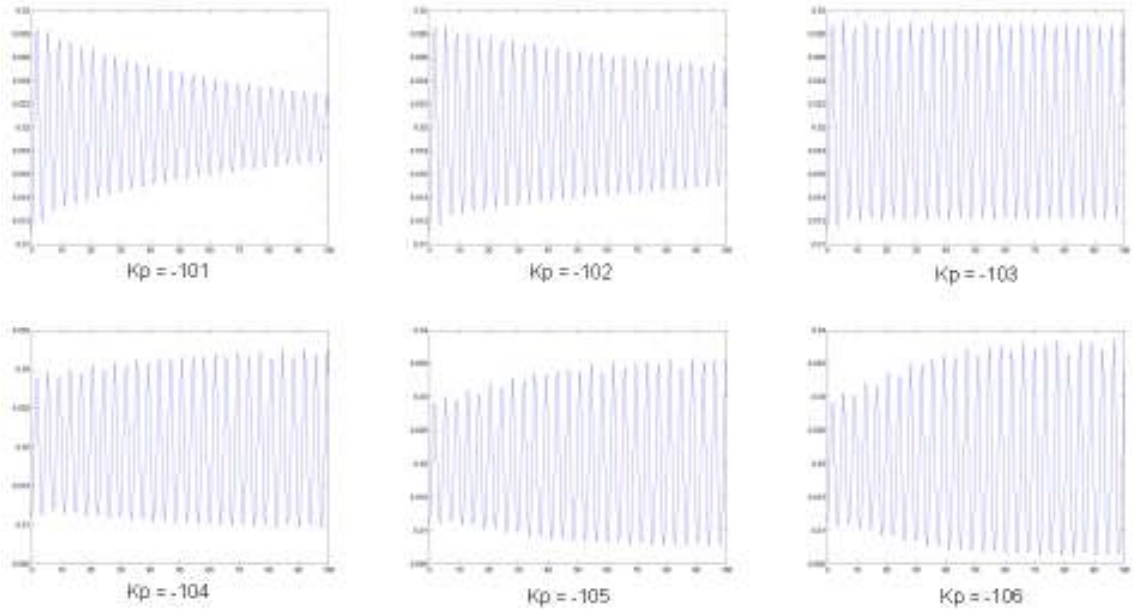
13. Appendix IV - Ziegler-Nichols experiments

13.1. Control loop in x_B

Results for the controlled variable x_B for values of K_p =[-10, -120]

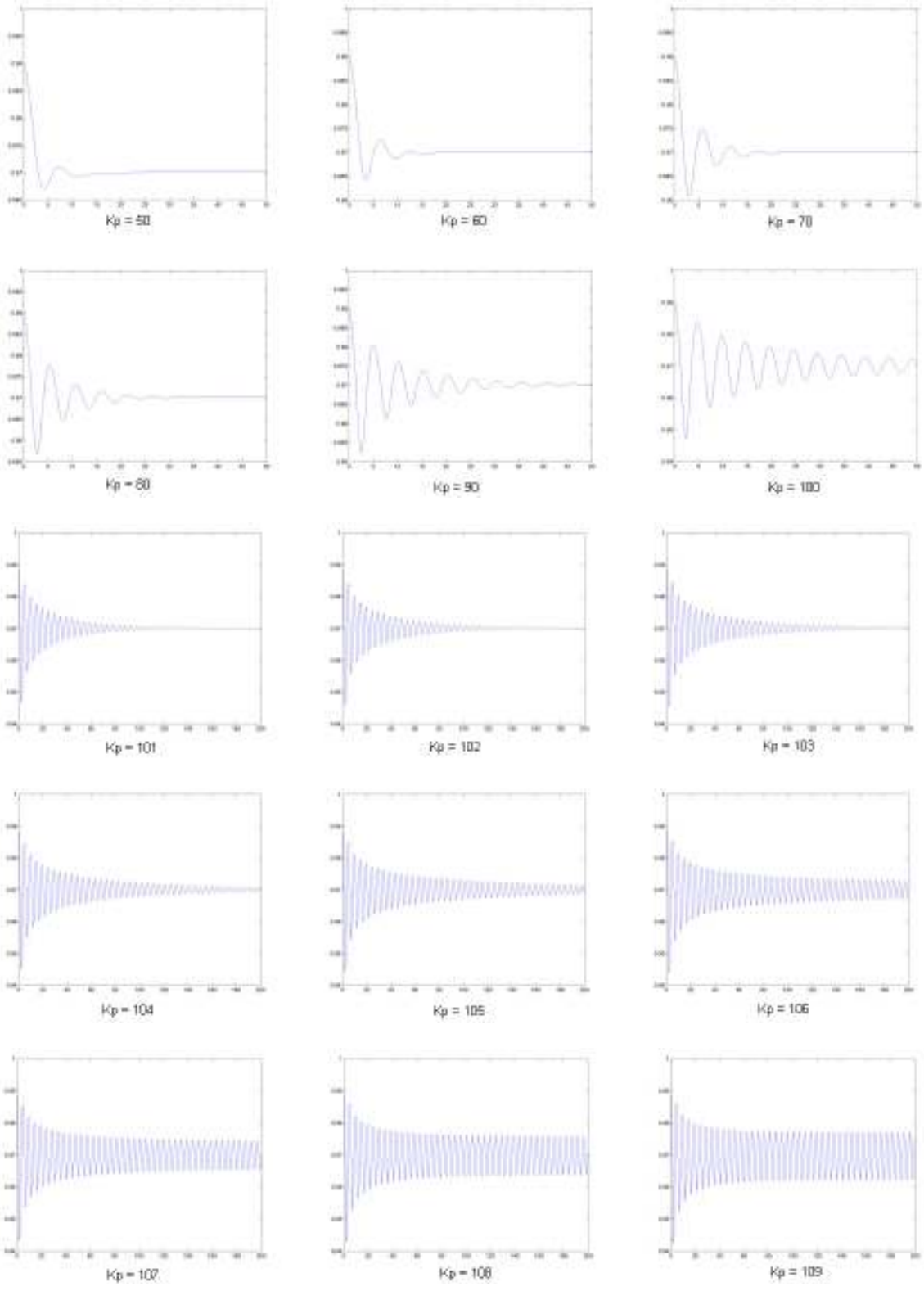


Results for the controlled variable x_B for values of K_p =[-101, -106]



13.2. Control loop in y_D

Results for the controlled variable y_D for values of K_p =[50, 210]



14. Appendix V - Adding secondary points

A potential concern in the use of two-level factorial designs is the assumption of linearity in the factor effects [8]. Of course, perfect linearity is unnecessary, and the 2^k system will work quite well even when the linearity assumption holds only approximately. However, there is a method of replicating certain points in the 2^k factorial that will provide protection against curvature as well as allow an independent estimate of error to be obtained. The method consists of adding center points to the 2^k design. These consist of n_C replicates run at the point $x_i = 0$ ($i = 1, 2, ..., k$). One important reason for adding the replicate runs at the design center is that center points do not affect the usual effects estimates in a 2^k design.

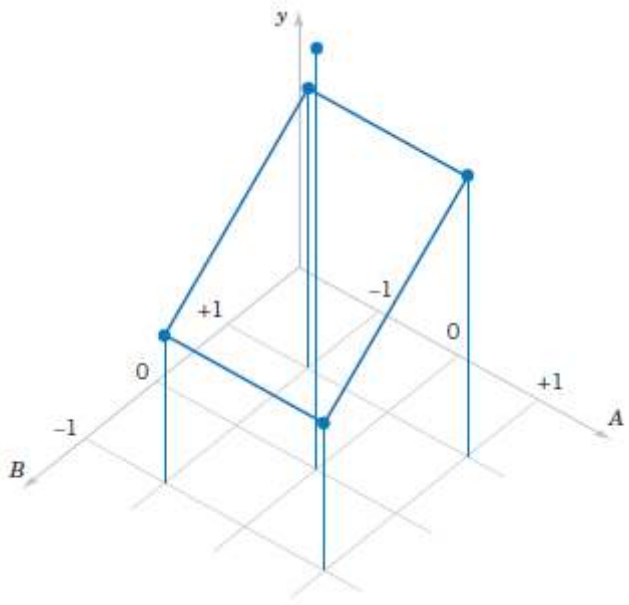


Figure 14.1.- Design with center points [8]

Consider a 2^2 design with one observation at each of the factorial points (-,-), (+, -), (-, +), and (+, +) and n_C observations at the center points (0, 0) (see Figure 14.1). Let $\bar{y}_F$ be the average of the four runs at the four factorial points, and let $\bar{y}_C$ be the average of the n_C run at the center point. If the difference $\bar{y}_F - \bar{y}_C$ is small, the center points lie on or near the plane passing through the factorial points, and there is no curvature. On the other hand, if $\bar{y}_F - \bar{y}_C$ is large, curvature is present. A single degree-of-freedom sum of squares for curvature is given by

$$SS_{Curvature} = \frac{n_F n_C \left(\bar{y}_F - \bar{y}_C\right)^2}{n_F + n_C} = \left(\frac{\bar{y}_F - \bar{y}_C}{\sqrt{\frac{1}{n_F} + \frac{1}{n_C}}} \right)^2 \qquad \textbf{Eq. 14.1}$$

where n_F is the number of factorial design points. This quantity may be compared to the error mean square to test for curvature.

When points are added to the center of the 2^k design, the model we may entertain is:

$$Y = \beta_0 + \sum_{j=1}^{k} \beta_j x_j + \sum\sum_{i<j} \beta_{ij} x_i x_j + \sum_{j=1}^{k} \beta_{jj} x_j^2 + \epsilon$$

where the β_{ij} are pure quadratic effects. The test for curvature actually tests the hypotheses

$$H_0: \sum_{j=1}^{k} \beta_{jj} = 0$$

$$H_1: \sum_{j=1}^{k} \beta_{jj} \neq 0$$

Furthermore, if the factorial points in the design are unreplicated, we may use the n_C center points to construct an estimate of error with n_C - *1* degrees of freedom.

There is another possibility with the addition of star points [9]. The *Box-Wilson Central Composite Design* (*CCD*) contains an embedded factorial or fractional factorial design with center points that is augmented with a group of *star points* that allow estimation of curvature. If the distance from the center of the design space to a factorial point is ±1 unit for each factor, the distance from the center of the design space to a star point is $\pm\alpha$ with $|\alpha| > 1$. The precise value of alpha depends on certain properties desired for the design and on the number of factors involved. If the factorial is a full factorial, then

$$\alpha = \left(2^k\right)^{1/4} \qquad \textbf{Eq. 14.2}$$

A *CCD* always contains twice as many star points as there are factors in the design. The star points represent new extreme values (low and high) for each factor in the design. Figure 14.2 illustrates the relationships among these varieties. Note that the *CCC* explores the largest process space and the *CCI* explores the smallest process space. In the *CCC* design, the design points describe a circle circumscribed about the factorial square. For three factors, the *CCC* design points describe a sphere around the factorial cube.

Table 14.1 summarizes the properties of three varieties of central composite designs. *CCC* designs provide high quality predictions over the entire design space, but require factor settings outside the range of the factors in the factorial part. *CCI* designs use only points within the factor ranges originally specified, but do not provide the same high quality prediction over the entire space compared to the *CCC*. *CCF* designs provide relatively high quality predictions over the entire design space and do not require using points outside the original factor range. However, they give poor precision for estimating pure quadratic coefficients.

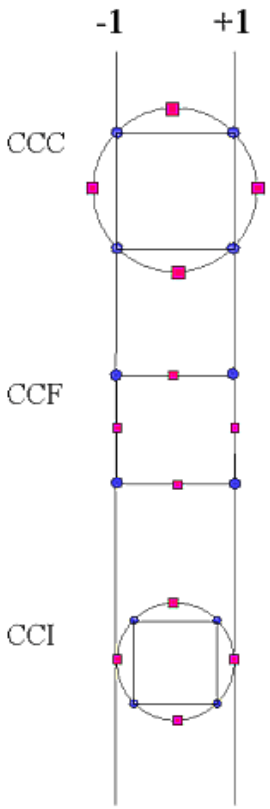


Figure 14.2.- Comparison of three types of CCD

Table 14.1.- Types of Central Composite Designs

Type and description
Circumscribed designs (*CCC*) are the original form of the central composite design. The star points are at some distance alpha from the center based on the properties desired for the design and the number of factors in the design. The star points establish new extremes for the low and high settings for all factors. These designs have circular, spherical, or hyperspherical symmetry and require 5 levels for each factor.
Face centered designs (*CCF*). In this design the star points are at the center of each face of the factorial space, so α = ± 1. This variety requires 3 levels of each factor.
Inscribed designs (*CCI*). For those situations in which the limits specified for factor settings are truly limits, the *CCI* design uses the factor settings as the star points and creates a factorial or fractional factorial design within those limits. In other words, a *CCI* design is a scaled down *CCC* design with each factor level of the *CCC* design divided by alpha to generate the *CCI* design. This design also requires 5 levels of each factor.

15. Appendix VI

Multiple response optimization results for both control loops (x_B and y_D) for several percentage variations of the initial Ziegler-Nichols parameters:

Optimized loop: xB
Variation:5
Step 1
R-squared(xB) = 0.986877; R-squared(yD) = 0.974090
Kp_xB = -42.182464; Ki_xB = 5.310914; Kp_yD = 45.674624; Ki_yD = 3.814765
ISE_xD = 0.001322 (-12.624047%); ISE_yD = 0.000209 (-0.407889%)
Step 2
R-squared(xB) = 0.862515; R-squared(yD) = 0.973638
Kp_xB = -40.627804; Ki_xB = 5.838548; Kp_yD = 41.703130; Ki_yD = 3.440065
ISE_xD = 0.001254 (-17.127023%); ISE_yD = 0.000211 (0.328976%)
********End of optimization********

Optimized loop: xB
Variation:10
Step 1
R-squared(xB) = 0.989651; R-squared(yD) = 0.997271
Kp_xB = -41.133826; Ki_xB = 5.765431; Kp_yD = 38.988690; Ki_yD = 3.219998
ISE_xD = 0.001254 (-18.030909%); ISE_yD = 0.000217 (3.555067%)
********End of optimization********

Optimized loop: xB
Variation:15
Step 1
R-squared(xB) = 0.973937; R-squared(yD) = 0.993373
Kp_xB = -43.466558; Ki_xB = 6.272105; Kp_yD = 38.059931; Ki_yD = 2.844313
ISE_xD = 0.001263 (-17.283541%); ISE_yD = 0.000207 (-1.031119%)
Step 2
R-squared(xB) = 0.963290; R-squared(yD) = 0.994885
Kp_xB = -43.076707; Ki_xB = 8.031884; Kp_yD = 42.021564; Ki_yD = 3.690339
ISE_xD = 0.001181 (-22.684447%); ISE_yD = 0.000210 (0.197466%)
********End of optimization********

Optimized loop: xB
Variation:20
Step 1
R-squared(xB) = 0.902170; R-squared(yD) = 0.976792
Kp_xB = -44.979464; Ki_xB = 6.760130; Kp_yD = 50.264385; Ki_yD = 3.097949
ISE_xD = 0.001309 (-13.706296%); ISE_yD = 0.000173 (-17.282709%)
Step 2
R-squared(xB) = 0.930855; R-squared(yD) = 0.988419
Kp_xB = -45.440935; Ki_xB = 9.336346; Kp_yD = 53.259297; Ki_yD = 4.332833
ISE_xD = 0.001219 (-19.666919%); ISE_yD = 0.000182 (-13.061919%)
********End of optimization********

Optimized loop: xB
Variation:25
Step 1
R-squared(xB) = 0.774632; R-squared(yD) = 0.853717
Kp_xB = -42.243254; Ki_xB = 7.215256; Kp_yD = 48.968477; Ki_yD = 3.297891
ISE_xD = 0.001264 (-17.168015%); ISE_yD = 0.000176 (-15.742013%)
Step 2
R-squared(xB) = 0.913947; R-squared(yD) = 0.984964
Kp_xB = -42.177546; Ki_xB = 9.841186; Kp_yD = 54.184089; Ki_yD = 4.941802
ISE_xD = 0.001225 (-19.718836%); ISE_yD = 0.000185 (-11.512917%)
********End of optimization********

Optimized loop: xB
Variation:30
Step 1
R-squared(xB) = 0.685863; R-squared(yD) = 0.641732
Kp_xB = -45.709211; Ki_xB = 5.653221; Kp_yD = 48.861951; Ki_yD = 3.358590
ISE_xD = 0.001381 (-10.047249%); ISE_yD = 0.000188 (-9.936679%)
Step 2
R-squared(xB) = 0.696328; R-squared(yD) = 0.658460
Kp_xB = -43.310701; Ki_xB = 8.524767; Kp_yD = 56.743094; Ki_yD = 4.158409
ISE_xD = 0.001258 (-18.064577%); ISE_yD = 0.000170 (-18.523056%)
Step 3
R-squared(xB) = 0.814393; R-squared(yD) = 0.956689
Kp_xB = -43.252833; Ki_xB = 11.941069; Kp_yD = 61.162707; Ki_yD = 6.649926
ISE_xD = 0.001240 (-19.219631%); ISE_yD = 0.000190 (-9.081702%)
********End of optimization********

Optimized loop: yD
Variation:5

Step 1
R-squared(xB) = 0.999984; R-squared(yD) = 0.999986
Kp_xB = -51.128276; Ki_xB = 4.486906; Kp_yD = 53.152262; Ki_yD = 3.645160
ISE_xB = 0.000209 (-1.142232%)
ISE_yD = 0.000105 (-10.167276%)
Step 2
R-squared(xB) = 0.999976; R-squared(yD) = 0.999980
Kp_xB = -53.280007; Ki_xB = 4.197596; Kp_yD = 53.522933; Ki_yD = 3.282252
ISE_xB = 0.000206 (-2.315807%)
ISE_yD = 0.000101 (-14.105872%)
Step 3
R-squared(xB) = 0.999981; R-squared(yD) = 0.999976
Kp_xB = -54.330760; Ki_xB = 3.975151; Kp_yD = 52.598368; Ki_yD = 3.013469
ISE_xB = 0.000204 (-3.572001%)
ISE_yD = 0.000099 (-15.593272%)
Step 4
R-squared(xB) = 0.999982; R-squared(yD) = 0.999969
Kp_xB = -54.390211; Ki_xB = 3.827485; Kp_yD = 52.581672; Ki_yD = 2.996695
ISE_xB = 0.000200 (-5.357134%)
ISE_yD = 0.000099 (-15.373002%)
********End of optimization********

Optimized loop: yD
Variation:10

Step 1
R-squared(xB) = 0.999754; R-squared(yD) = 0.999741

```
        Kp_xB = -48.753961;  Ki_xB = 3.953895;  Kp_yD = 49.915681;  Ki_yD = 3.253482
        ISE_xB = 0.000210 (-0.613015%)
        ISE_yD = 0.000106 (-9.610620%)
Step 2
        R-squared(xB) = 0.999760;  R-squared(yD) = 0.999628
        Kp_xB = -52.551290;  Ki_xB = 3.322552;  Kp_yD = 50.527873;  Ki_yD = 2.626936
        ISE_xB = 0.000209 (-0.896360%)
        ISE_yD = 0.000098 (-16.263358%)
Step 3
        R-squared(xB) = 0.999721;  R-squared(yD) = 0.999377
        Kp_xB = -52.966162;  Ki_xB = 3.260707;  Kp_yD = 50.057623;  Ki_yD = 2.567761
        ISE_xB = 0.000207 (-2.117613%)
        ISE_yD = 0.000098 (-16.218899%)
********End of optimization********

Optimized loop: yD
Variation:15

Step 1
        R-squared(xB) = 0.998387;  R-squared(yD) = 0.998098
        Kp_xB = -48.765438;  Ki_xB = 3.445371;  Kp_yD = 48.776085;  Ki_yD = 2.845332
        ISE_xB = 0.000211 (-0.035959%)
        ISE_yD = 0.000103 (-12.026724%)
Step 2
        R-squared(xB) = 0.998364;  R-squared(yD) = 0.997579
        Kp_xB = -51.762996;  Ki_xB = 2.841199;  Kp_yD = 46.623018;  Ki_yD = 2.242214
        ISE_xB = 0.000209 (-0.888083%)
        ISE_yD = 0.000100 (-14.838281%)
Step 3
        R-squared(xB) = 0.997586;  R-squared(yD) = 0.995280
        Kp_xB = -51.844855;  Ki_xB = 3.057757;  Kp_yD = 46.642640;  Ki_yD = 2.321068
        ISE_xB = 0.000207 (-1.788773%)
        ISE_yD = 0.000100 (-14.778469%)
********End of optimization********

Optimized loop: yD
Variation:20

Step 1
        R-squared(xB) = 0.994891;  R-squared(yD) = 0.994663
        Kp_xB = -65.529670
        Ki_xB = 4.303516
        Kp_yD = 54.540496
        Ki_yD = 2.438176
        ISE_xB = = 0.000202 (-4.344551%)
        ISE_yD = = 0.000089 (-23.929038%)
Step 2
        R-squared(xB) = 0.994645;  R-squared(yD) = 0.988320
        Kp_xB = -65.530872
        Ki_xB = 4.297720
        Kp_yD = 54.539223
        Ki_yD = 2.461615
        ISE_xB = = 0.000200 (-5.270528%)
        ISE_yD = = 0.000089 (-22.676787%)
********End of optimization********


Optimized loop: yD
Variation:25
```

Step 1
R-squared(xB) = 0.984157; R-squared(yD) = 0.985560
Kp_xB = -52.202639; Ki_xB = 2.522390; Kp_yD = 50.643285; Ki_yD = 2.553763
ISE_xB = 0.000200 (-5.257169%)
ISE_yD = 0.000102 (-13.272091%)
Step 2
R-squared(xB) = 0.971792; R-squared(yD) = 0.952996
Kp_xB = -52.248477; Ki_xB = 3.131629; Kp_yD = 50.643961; Ki_yD = 2.700203
ISE_xB = 0.000201 (-4.884623%)
ISE_yD = 0.000100 (-14.742131%)
********End of optimization********

Optimized loop: yD
Variation:30

Step 1
R-squared(xB) = 0.957197; R-squared(yD) = 0.969284
Kp_xB = -52.823405; Ki_xB = 2.073271; Kp_yD = 52.526331; Ki_yD = 2.538489
ISE_xB = 0.000217 (2.958138%)
ISE_yD = 0.000104 (-10.863982%)
********End of optimization********

16. References

1. Skogestad, S. "MATLAB Distillation Column Model ("Column A")." *Distillation: Column A*. Web.10 Sept. 2012. <http://www.nt.ntnu.no/users/skoge/book/matlab_m/cola/cola.html>.
2. "Simulink." *Simulink - Simulation and Model-Based Design*. MathWorks®. Web. 10 Sept. 2012. <http://www.mathworks.com/products/simulink/index.html>.
3. Kister, Henry Z. "Chapter 1." *Distillation Design*. New York: McGraw-Hill, 1992. 3-14.
4. Smith, J. M., Hendrick C. Van. Ness, and Michael M. Abbott. *Introduction to Chemical Engineering Thermodynamics*. 7th ed. Boston: McGraw-Hill, 2005.
5. Treybal, Robert E. "Chapter 9." *Mass Transfer Operations*. Third ed. New York: McGraw-Hill, 1981. 402-15.
6. Skogestad, S. "Dynamics and Control of Distillation Columns. A Tutorial Introduction." *Chemical Engineering Research and Design* 75.6 (1997): 539-62
7. S. Skogestad, M. Morari. "The dominant time constant for distillation columns" *Computers & Chemical Engineering* 11.6 (1987): 607-17
8. Montgomery, Douglas C., and George C. Runger. "Chapter 14." *Applied Statistics and Probability for Engineers*. Third ed. New York: Wiley, 2003. 9-20
9. "Central Composite Designs (CCD)." *5.3.3.6.1. Central Composite Designs (CCD)*. Web. 10 Sept. 2012. <http://www.itl.nist.gov/div898/handbook/pri/section3/pri3361.htm>.
10. Seborg, Dale E., Thomas F. Edgar, and Duncan A. Mellichamp. "Chapter 1." *Process Dynamics and Control*. Hoboken, NJ: Wiley, 2004. 1-7.
11. Stephanopoulos, George. "Chapter 2." *Chemical Process Control: An Introduction to Theory and Practice*. New Delhi: Prentice-Hall of India, 1998.
12. Seborg, Dale E., Thomas F. Edgar, and Duncan A. Mellichamp. "Chapter 8." *Process Dynamics and Control*. Hoboken, NJ: Wiley, 2004.
13. Kister, Henry Z. "Chapter 18." *Distillation Operations*. New York: McGraw-Hill, 1990.
14. Kano, M., Miyazaki, K., Hasebe, S., Hashimoto, I. (2000). *Inferential control system of distillation compositions using dynamic partial least squares regression*. Journal of Process Control, 10:157–166.
15. Martens, H., Naes, T. (1989). Multivariate Calibration. New York: Wiley.
16. Stephanopoulos, George. "Chapter 16." *Chemical Process Control: An Introduction to Theory and Practice*. New Delhi: Prentice-Hall of India, 1998:310
17. Skogestad, Sigurd, and Ian Postlethwaite. "Chapter 2." *Multivariable Feedback Control: Analysis and Design*. Hoboken, NJ: John Wiley, 2005. 39.

18. Luyben, Michael L., and William L. Luyben. "Chapter 3." *Essentials of Process Control*. New York: McGraw-Hill, 1997. 96-98.

19. “Single Response: Path of Steepest Ascent." *5.5.3.1.1. Single Response: Path of Steepest Ascent*. Web. 10 Sept. 2012. <http://www.itl.nist.gov/div898/handbook/pri/section5/pri5311.htm>.

20. Jim, Riggs. "Distillation: Major Disturbances & First-Level Control." *Distillation: Major Disturbances & First-Level Control*. N.p., n.d. Web. 10 Sept. 2012. <http://controlstation.brightegg.com/page/203-distillation-major-disturbances-and-first-level-control>.

21. MacGregor, J. F., and T. Kourti. "Statistical Process Control of Multivariate Processes." *Control Engineering Practice* 3.3 (1995): 403-14.

22. Ferrer, Alberto. "Multivariate Statistical Process Control Based on Principal Component Analysis (MSPC-PCA): Some Reflections and a Case Study in an Autobody Assembly Process." *Quality Engineering* 19.4 (2007): 311-25.

23. Jackson, J. E. (2003). *A User´s Guide to Principal Components*. New York: Wiley

24. 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.

25. Martens, H., Naes, T. (1989). *Multivariate Calibration*. New York: Wiley.

26. Woodall, W.H. (2000). *Controversies and contradictions in statistical process control*. Journal of Quality Technology, 32(4):341–350.

27. Wold, H. (1966). *Nonlinear estimation by iterative least squares procedures*. In: David, F. Ed. Research Papers in Statistics. New York:Wiley, 411–444.

28. Wold, S. (1978). *Cross-validatory estimation of the number of components in factor and principal component models*. Technometrics, 20(4):397–405.

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

30. Multi- and Megavariate data analysis. Part I (Umetrics)

31. Tracy, N. D., Young, J. C., Mason, R. L. (1992). Multivariate control charts for individual observations. Journal of Quality Technology, 24(2):88–95.

32. Kourti, T., MacGregor, J. F. (1996). Multivariate SPC methods for process and product monitoring. Journal of Quality Technology, 28(4):409–428.

33. Box, G. E. P. (1954). *Some theorems on quadratic forms applied in the study of analysis of variance problems: effect of inequality of variance in Oneway classification*. The Annals of Mathematical Statistics, 25:290–302

34. Nomikos, P., MacGregor, J. F. (1995). *Multivariate SPC charts for monitoring batch processes*. Technometrics, 37(1):41–59.

35. Luyben, William L. "Control of CSTR Systems." Chemical Reactor Design and Control. [New York]: AIChE, 2007

36. Harris, T. *A Review of Performance Monitoring and Assessment Techniques for Univariate and Multivariate Control Systems*. Journal of Process Control 9.1 (1999): 1-17.

37. Qin, S. J. *Control Performance Monitoring - A Review and Assessment*. Comput. Chem. Eng. 1998, 23, 173.

38. T. O'MAHONY, C.J. DOWNING, K. FATLA: *Genetic algorithms for PID parameter optimisation: minimising error criteria*. Process Control and Instrumentation 2000: 148 - 53.

39. J. M. Herrero, X. Blasco, M. Martínez, and J. V. Salcedo. *Optimal PID tuning with genetic algorithms for non linear process models*. IFAC 15th Triennial World Congress, 2002.

40. Herreros, A., E. Baeyens and J.R. Perán (2000). *Design of PID controllers using multiobjective genetic algorithms*. Workshop on Digital control: Past, present and future of PID Control, PID'00.

41. Kourti, T., *Multivariate dynamic data modeling for analysis and Statistical process control of batch processes, start-ups and grade transitions*, Journal of Chemometrics (2003) 17: 93-109

42. Camacho, E. F., and C. Bordons. *Model Predictive Control*. Berlin: Springer, 1999.

43. S. Joe Qin, Thomas A. Badgwell, *A survey of industrial model predictive control technology*, Control Engineering Practice, 11 (2003): 733–64.