UNIVERSITÀ DEGLI STUDI DI PISA

Dipartimento di Ingegneria Chimica,

Chimica Industriale e Scienza dei Materiali

Tesi di Dottorato in Ingegneria Chimica

Sviluppo di tecniche di monitoraggio delle prestazioni di processi chimici controllati

Relatori:
Prof. Ing. CLaudio Scali

Candidato:
Rossi Maurizio

Ciclo **XVIII**

UNIVERSITY
OF PISA

Department of Chemical Engineering, Industrial Chemistry and Material Science

Chemical Process Control Laboratory (CPCLab)

Development of Techniques for Closed Loop Performance Monitoring of Chemical Processes

Rossi Maurizio

Degree of Doctor of Philosophy in Chemical Engineering
2007

Ho un rapporto catastrofico con la tecnologia:
se passo sotto ad un lampadario a gocce, si mette a piovere.

Woody Allen

Acknowledgment

During my studies I have met a lot of people from different countries, with different culture and, as consequence, different point of view about life. I want to thank everyone for helping me in opening my mind and sharing with me always quality time.

First of all, my sincere thanks go to Claudio Scali, professor and friend; with him, I want to name all my colleagues in Pisa: Raffaella, Gianfilippo, Jacopo, Fabio, Serena, Rachele, Andrea and Lorenzo. Of course I cannot forgive the two Gabriele: Pannocchia (thanks for helping me with network problems at CPClab and many other issues) and Marchetti (thanks for inviting me to join Aspentech and teaching me how to use a check-list!).

Among my friends, I want to give a special thank to Paolo (otherwise called *Paolone*, the reader will certainly imagine the reason) and Melania (we surely will not forget both of you, Aosta is only one mile far in our heart!), Francesca and Alessandro (dinners and smiles in your home are always present). Thanks guys to be here with me, I do not have enough word to describe my feelings for you.

Also, I want to name my little cousin Barbara (with Massimiliano and MariaPia) for not being tired to invite me continually to have a lunch together; it's only a matter of time, I assure you.

A special thank also to my cousin Silvia, who has shared a lot of problems with me, listening with patience, especially in front of a good lunch!

During 2005 I had the opportunity of being employed in Huntsman-Tioxide as a process engineer; there I found an unexpected and truly well accepted welcome; my thanks to Mariagrazia, Melania, Chiara, Domenico, Francesco, Omar, Roberto, Nausica, Livio, Eduardo, Anna (Giuntoli), Gianluca, Marta, Diego, Patrizio, Emilio, Ivosco, Mario, Antonio, Marco (Caramelli), Bruno, Stefano, Andrea, Massimo (Ciampoli), Franco, Anna (Montemaggi), Vincenzo, Luca, Gianni, Nilo, Massimo (Cascioli), Marco (Cerquettini), Paolo, (Ales)Sandro, Marcello, Simone, Leonetto, Alessandro, Cristiana, Sergio and all the management staff, all operators, everyone.

Particularly I wish to thank Mariagrazia, my ex-boss now and Gianluca, my ex-

ex boss: we shared *a series of unfortunate events*, overcame by our friendship (you will surely understand).

With Tioxide I had the possibility to go in Yarm in England where I met people which now I call friend: Ian and Justin from Greatham (U.K.); Kerry (from Texas, a very skilled man); Ilaria, Mafalda and Jean-Marc (from Switzerland); Mathew, Helen and Eric (from France) and many others. Particularly, I feel the need of thanking Ilaria, Mafalda and Helen for trying to teach me some French (with very scarce results, I should admit).

During my studies I spent seven months in Canada, at the University of Alberta in Edmonton. There I met Professor Sirish L. Shah and his wife: I still keep in my mind the dinners we had and the time we passed together; with their effort I forgot my homesick for Italy. My Canadian friends deserve here to be remembered: Paul and Lourie (I still remember the heart-attacks after all ours *short* jogging in Edmonton parks), Amin, Leith, Shoukat, Arun, Rumana, Huang, Hijing and Li, and all the other people I met in the campus.

Now, I'm working for Aspentech, and for this reason I've met new (and old) friends: Filippo and Michela (and their little Ludovica, the real boss in Pisa's office), prof. Alessandro Brambilla, Alessandro (il *Masio*), Alessandro (il *Trippi*), Gianpaolo, Giancarlo and Irene, Gabriele (Marchetti, above cited) and Giulia, Cristiano and Roger (actually in Reading, UK).

Finally the most important people which are part of my life: my parents Luciano e Mariarosa (they were always on my side, pushing and supporting me to reach my objectives); my grandmother Giorgia (the most significant gift is her presence in my life); my parents-in-law (yes, I finally got married!) Stefano and Valeria; my brother-in-law Matteo and his girlfriend Karina.

And now the two persons which are not a part of my life but the *entire* life, Veronica and our little creature Matilde, in her way to start a new life in this world: thanks for being here.

To my little Matilde,
who will bring a new light in my life

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 General issues on control loop performance

A chemical plant nowadays presents a large number of controllers, which help the operator to maintain the process on target, thus obtaining savings and benefits in terms of costs and quality. With the increase in the number of automatic systems in a plant the task of assessing their performance becomes of vital importance.

Even though differences can be found among all applications, a good monitoring tool should have some important characteristics: simplicity and friendly interaction with the operator, short calculation times, flexibility in applications both on-line and off-line, compatibility with industrial software, low (possibly none) additional perturbations introduced in the system to perform the diagnosis, thus basing the analysis on data recorded during routine operation. Data must be collected from the plant and stored on hard drives for off-line techniques, or processed directly with on-line techniques. The aim of the former approach is to describe the status of the plant at the time of the data collection: larger data set can be handled and complexity of techniques is not problematic, but the compression of data usually adopted represents a strong drawback. In the latter approach the advantage is an immediate indication of poor performing loops, but often is counterbalanced by limitations as well as network speed or DCS (Distributed Control System) load which require the adoption of techniques based on simple calculations.

Whatever the approach, results must be rapidly and easily understood by operators: indexes are often used and compared with some limit values which indicate the passage from good to poor performing loops and viceversa.

1.2 Monitor of Performance and Diagnosis

Performance monitoring in chemical plant data has received a larger importance recently, with the aim of controlling quality of products and economic yield of the process, as witnessed by the interest both from academic research and from advanced industrial applications ([2], [3]).

A control loop can be considered problematic for different causes: from uncorrect design or tuning of controllers, to sensor failures, to the presence of friction in actuators. Also, perturbations entering in the plant may affect the performance of the loop. The detection of the cause allows the operator to intervene either on the control system or on the plant to counterbalance the problem.

The issue, which has been studied for a longer time, is certainly about uncorrect tuning of controllers: either due to initial errors generated by excessive prudence to avoid oscillatory behavior or changes in parameters, due to changes in operating conditions. Autotuning techniques are wildly diffused in industrial applications ([4], [5]). A more frequent problem is the presence of static friction in actuators, phenomenon which generates delays in actuation of the control action, with the possibility of having the valve completely stuck ([6]). Static friction effect is an oscillation on controlled variable, similarly to aggressive controllers or sinusoidal disturbances entering in the plant.

From these considerations becomes clear the importance of a monitoring system able to distinguish among causes and to suggest opportune countermeasures.

1.3 Thesis Overview

In a first part of this thesis different techniques, already presented in literature, are discussed and in some case modified to increase their applicability on different sets of data. Furthermore, new techniques both for monitoring and diagnosis problems are described. All of them are organized in a software tool which can be used for an off-line analysis. The main objective of the described techniques is to label loops and to concentrate the attention of the operator on those which requires adjustments.

As described in **Chapter 2**, the detection of poor performing loops is achieved as first step of the whole procedure: the output of this phase is a initial classification among *good performing*, *oscillating* and *sluggish* loops. Oscillating loops are therefore analyzed in more detail to determine the cause of oscillations. The presence of pure noise is not of interest and the following analysis are done only on other cases. Through the application of techniques in time domain, loops affected by significative oscillations in controlled variables or sluggish disturbance rejection are detected. Hence, using frequency domain techniques,oscillating loops are classified more in detail.

At this point, some techniques can be applied to diagnose the cause of the poor performance on oscillating loops distinguishing among static friction on actuators, marginal stability of controllers or external disturbances. Eventually, thanks to this classification, the proper countermeasure can be decided: a stiction index can be evaluated or a new set of tuning parameters calculated. Two chapters belongs to this section:

- **Chapter 3**: it reports a detailed description of static friction models with two approaches (data driven models, and model based on a balance of forces on actuators); the effect of stiction on collected data is analyzed and several techniques for its detection are described and compared; finally a stiction index is proposed to indicate the level of static friction.
- **Chapter 4**: here the problem of poor performing controllers is considered, dividing the procedure in three steps: identification, retuning and performance evaluation. Two techniques are described to find models of process and disturbances, used in the sequel to obtain new sets of controller parameters by adopting several tuning methodologies; also, some performance indexes are defined.

The following chapter (**Chapter 5**) is entirely dedicated to the development of the software tool designed to organize in a single application all the techniques described previously. Finally, a first approach to the $MIMO$ (Multiple Input Multiple Output) problem will be described and a procedure to detect the magnitude of interactions among loops will be analyzed (**Chapter 6**). Conclusions and main contributes of this thesis are summarized in **Chapter 7**; further developments are also discussed.

Chapter 2

Loop Classification

The $SISO$ (Single Input Single Output) will be here considered; the reference scheme depicted in figure 4.1 shows its principal elements: the controller **C**, process **P** and disturbance **Pd** dynamics, and the valve **V**. Generally set-point (SP), controller output (OP) and process variable (PV) are collected and stored on databases; on the contrary the collection of manipulated variables (MV) is not always performed. Other information, which could be useful for the loop analysis are both the sampling time T_s and the controller parameters, namely its gain K_C, integral time constant τ_I, derivative time constant τ_D and filter constant F_C.

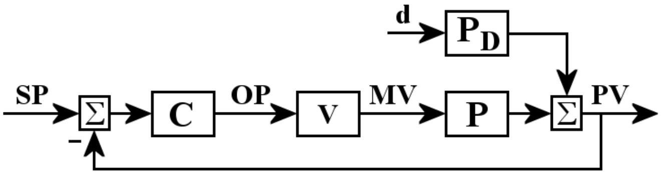


Figure 2.1: *Reference scheme for a Single Input Single Output ($SISO$) system*

This chapter describes different techniques which labels loops on the basis of collected trend dividing them into three categories: good performing loops, which are affected only by plant noise and consequently do not require further analysis; presence of a sluggish disturbance rejection due to an improper controller action; presence of oscillations due to different problems, not recognizable in this phase. The second step goes into a more subtle classification among oscillating loops. Techniques in frequency domain and in time domain are applied to find damped oscillations, loops with one predominant frequency, loops with a mixture of different frequency channels with similar power. The proposed classification is propaedeutic for the application of more complex techniques, described in following chapters,

*Portions of this chapter are published in [7], [8], [9] which are used to suggest the proper counteraction to eliminate the source of poor performance.

2.1 Identification of oscillating loops

2.1.1 Hägglund technique

Different techniques to detect oscillations in loops are proposed in literature, among them: Hägglund [1] proposes a technique based on the absolute value of errors $e = SP - PV$, Thornhill [10] uses the previous approach together with the application of the Harris index [11] and in [12] describes a power spectrum based technique to detect recurrent frequency channels in signals.

Techniques in the frequency domain were adopted in a second step of this analysis, hence in this first phase the Hägglund technique [1] was considered. Referring to [1], the term *oscillation* will be used to indicate the error *e*, lying between two consecutive crossing zero (figure 2.2).

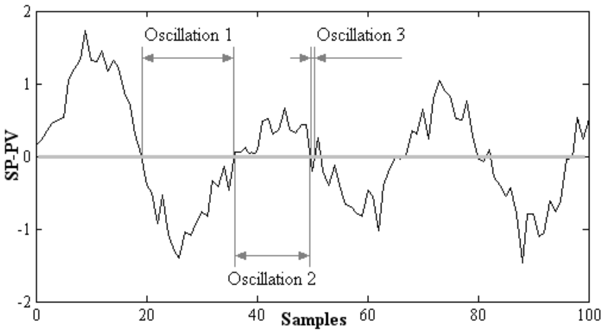


Figure 2.2: *Examples of definition for term* oscillation*, as described in [1]*

As basic assumption, the presence of anomalies defined as a significant oscillations in opposition to oscillations due only to noise, should be revealed wherever a *high* or *persistent* error is detected. This approach clearly indicates two cause which generates poor performance: not only the amplitude of errors should be taken into account, but also its spreading in time, thus avoiding random factors whose error continuously crosses the zero value, as well as plant noise. To summarize these two aspects in one function of the error, a modified Integral of Absolute Error (IAE_{Mod}) is suggested. The integral is evaluated every crossing zero of the error:

$$IAE_{mod} = \int_{t_i}^{t_{i+1}} |e(t)|dt \qquad \text{(Eq. 3-1)}$$

where t_i, t_{i+1} are times of two consecutive zero crossing of $e(t)$. The extension of Eq. 3-1 for sampled signals is:

$$IAE_{Mod}(t_k) = IAE_{Mod}(t_{k-1}) + |e(t_k)| \cdot Ts \qquad \text{(Eq. 3-2)}$$

where Ts is the sampling time. When the signal e(t) shows a zero crossing $IAE_{Mod}(t_k)$ is set again equal to zero instead of being calculated by Eq. 3-2. The result is a trend composed by a sequence of peaks; the larger the error or the longer its duration before next crossing zero, the higher the maximum of the peak, as shown in figure 2.3.

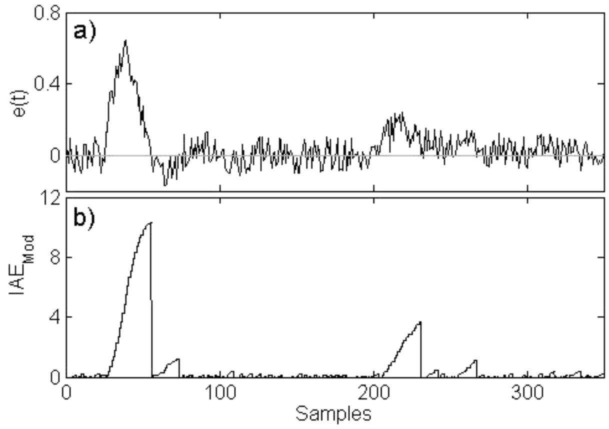


Figure 2.3: *Application of the Hägglund algorithm: a) error e(t); b) IAE_{Mod} peaks*

These integrated deviations are compared to a threshold value, named IAE_{Lim}, which represent the integral of an half-cycle sinusoidal oscillation with well defined amplitude (A_{sin}) and frequency (ω_{sin}):

$$IAE_{lim} = \int_0^{\pi/\omega_{sin}} A_{sin} \cdot (sin(\omega_{sin} \cdot t))dt = \frac{a}{\omega_{sin}} \qquad \text{(Eq. 3-3)}$$

If IAE_{Mod} overcomes IAE_{Lim} an anomaly is detected; Hägglund suggestion on the two parameters used in IAE_{lim} equation is: $A_{sin} = 1\%$ of the control range of the loop and $\omega_{sin} = \omega_U$ where ω_U is the ultimate frequency of the process. The latter information is not always available, but an approximation of ω_U value can be adopted: $\omega_U = 2 \cdot \pi/\tau_i$ (Hägglund [1]). This assumption is true in case of PI controllers tuned with Ziegler-Nichols technique [13], whose integral constant is strongly related to the ultimate period P_U of the process ($\tau_i = P_U/1.2$) and consequently to the ultimate frequency ($\omega_U = 2 \cdot \pi/P_U$). For other tuning techniques an error is introduced in the application of the algorithm. Henceforward in case of lack of knowledge on the ultimate frequency of the system it is suggested to choose directly the IAE_{Lim} value on the basis of historical data: the operator can distinguish between significant or negligible oscillations for each loop, thus indicating a more suitable value for the process under examination.

To check the presence of oscillations in a supervision time T_{Sup} a counter *x* is evaluated for each sample:

$$x = x \cdot \gamma + load \qquad \text{(Eq. 3-4)}$$

where *load*, generally equal to zero, becomes one for the sample in which an anomaly is detected ($IAE_{Mod} > IAE_{Lim}$); γ is a corrective factor:

$$\frac{1}{1-\gamma} = \frac{T_{Sup}}{Ts} \qquad \text{(Eq. 3-5)}$$

If x overcomes a limit value N_{Osc}, the loop is labelled *oscillating*. Hägglund proposes $N_{Osc} = 10$ in $T_{Sup} = 50 \cdot P_U \simeq 50 \cdot \tau_i$: those values, together with A_{Sin} and ω_{Sin} values, define completely the algorithm.

The most important advantage in adopting this technique resides in its simplicity: the use of simple formulas makes the algorithm easy to be translated in all the languages generally adopted on DCS (Distributed Control System). Furthermore the rapidity in analyzing data justifies both on-line and off-line applications. The use of ω_U of the process in estimating parameters of the algorithm constitutes the main drawback: the possibility of detecting low frequency oscillations ($\omega < \omega_U$) is precluded, as shown in the following.

2.1.2 Modified Hägglund technique

Generally, when a change in the process occurs and the controller starts to work in different operating conditions, an oscillation with frequency equal to ω_U could be generated (marginal stability). For those cases Hägglund technique [1] is able to detect correctly the presence of a significant oscillation. However, as described more in detail in the sequel, when a non-linear phenomenon is generating an oscillation (as well as static friction on actuators), the resulting frequency is completely uncorrelated with the ultimate frequency of the process. Hägglund technique is again applicable in case of frequencies bigger than ω_U but, when the frequency is lower, the technique fails in recognizing a significant oscillation.

For instance data from a refinery plant can be analyzed; $e(t)$ and $OP(t)$ trends are shown in figure 2.4:
The error $e(t)$ is clearly oscillating and Hägglund technique should recognize the presence of a significant perturbation. The integral constant of the controller was set equal to 12min and the control range between 0 and 25: with those data, $IAE_{Lim} =$

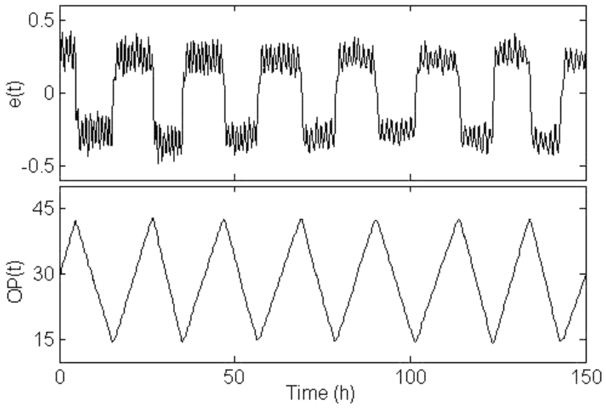


Figure 2.4: *Error e(t) and control action u(t)in case of oscillating loop*

0.95 and $T_{Sup} = 600min = 10h$. Looking at e(t) trend in figure 2.4, it is evident that whatever the IAE_{Lim} value, only a maximum of two oscillations can be detected in the supervision time T_{Sup} and the algorithm output indicates wrongly a *not oscillating* loop. The reason must be searched in a very low frequency of the oscillation ($\omega_{Osc} \simeq \omega_U/60$).

To solve this problem the original algorithm was modified eliminating the fixed supervision time in favor of a time moving window (see figure 2.5).

The basic idea is that with oscillating loops every half-cycle is repeated with similar duration. Hence, the new algorithm considers a window which starts from the ending point of the last collected anomalous half-cycle, whose duration is used to evaluate a supervision time (horizon of the window T_{Osc}).

Firstly, initialization of variables is made by performing three steps: i) waiting for the first crossing zero after starting the whole procedure; ii) assigning a zero value to the number of significant anomalies detected; iii) assigning an infinity value to the horizon of the time window T_{Sup}. Last two values will be updated in following steps of the algorithm.

The core of the process starts collecting the actual time value in a variable $T1$, representing the starting point of the anomaly under exam. Until IAE_{Mod}, evaluated for each sample, remains under IAE_{Lim} the algorithm is stuck in a inner cycle, being T_{Sup} not updated and maintaining an infinity value. To exit on the outer cycle only a significant oscillation must be detected: if the case, once a new crossing zero is encountered, the actual time value is stored in a variable $T2$, representing the finishing point of the examined anomaly. In following steps T_{Sup} and N_{Osc} values are updated:

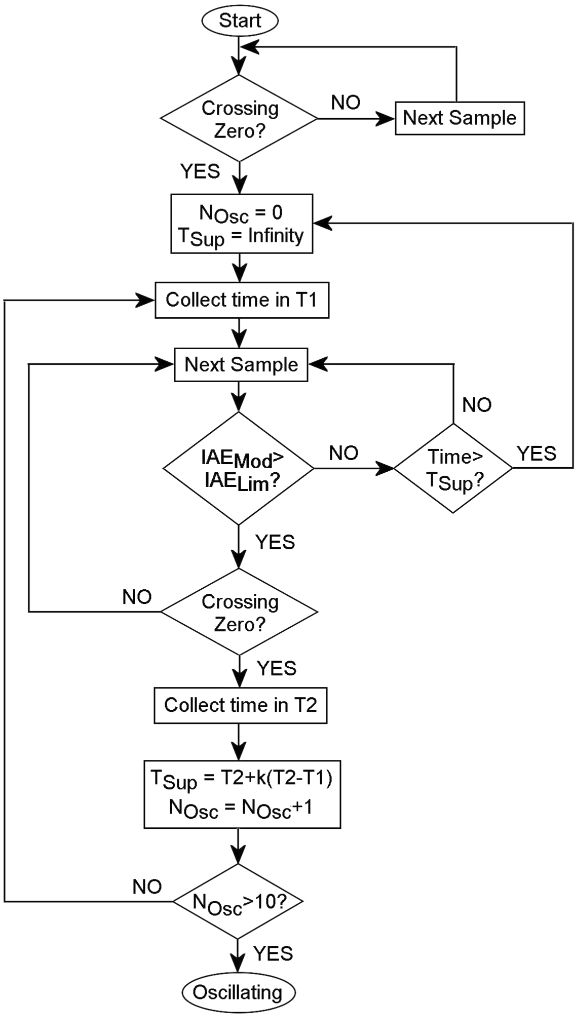


Figure 2.5: *Algorithm of the modified Hägglund technique*

$$T_{Sup} = T2 \cdot k(T2 - T1) \qquad \text{(Eq. 3-6)}$$

$$N_{Osc} = N_{Osc} + 1 \qquad \text{(Eq. 3-7)}$$

where k is set equal to 2 by default, meaning that the time moving window starts from $T2$ and is spread for 2 half-periods of the last significant oscillation.

Until the loop is not labelled as oscillating ($N_{Sup} > 10$), the algorithm processes an outer cycle, updating $T1$ value and proceeding as described before. In case no other anomalies are detected in the time moving window, all parameters are again initialized. This procedure, described for an on-line application, can be easily adapted also for off-line software: in the latter case the algorithm reaches an end either after processing all collected data or finding an oscillating behavior.

An example is reported for an oscillating loop in figure 2.6, where the suffix $^{(i)}$ is used to indicate the i^{th} application of the described algorithm. The first application reveals an anomaly (significant oscillation #1); henceforward the time moving window starts from $T2^{(1)}$ and finishes in $T^{(1)}_{Sup}$; N_{Osc} assumes unitary value. In a second step, the system finds a second anomaly (significant oscillation #2), belonging to the calculated time window: the window is shifted in the range $[T2^{(2)} \div T^{(2)}_{Sup}]$; N_{Osc} becomes equal to two. Time window range is equal in each case to two times $a^{(i)}$, because of the choice $k = 2$.

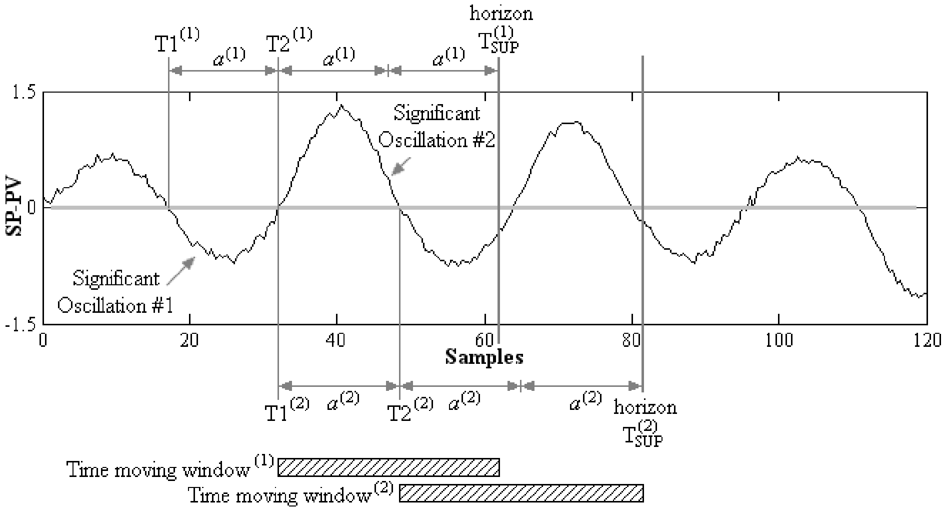


Figure 2.6: Example of time moving window for the modified Hägglund technique; two applications of the algorithm are shown, as indicated by suffixes $^{(1)}$ and $^{(2)}$

2.1.3 Application of modified Hägglund technique

In this subsection, the application of the modified Hägglund technique on eight loops from a refinery plant is analyzed; in table 2.1 controller parameters (gain K_C, integral constant τ_I, derivative constant τ_D and filter constant F_C), sampling time T_S, ultimate frequency approximated value $\hat{\omega}_U$ and control limits (lower $CTRL_L$, upper $CTRL_U$) are reported.

Data sets is composed by flow control loops (1, 2, 4 and 8), by pressure control loops (3 and 5) and a temperature control loop (7). Correspondent trends of PV (in black) and SP (in gray) are depicted in figure 2.7.

Loop	1	2	3	4	5	6	7	8
K_C	2	0.5	0.93	0.25	8	3.16	0.5	0.5
τ_I	30	12	90	90	120	50	180	12
τ_D	0	0	18	0	3	0	60	0
F_C	0.05	0.3	0.1	0.07	0	0	0	0.3
T_S	10	10	10	10	10	10	10	10
$\hat{\omega}_U = \tau_I/2\pi$	0.209	0.523	0.07	0.07	0.052	0.1256	0.035	0.52
$CRTL_L$	0	0	0	0	0	0	0	0
$CRTL_U$	110	25	80	1600	40	50	200	25

Table 2.1: *Values of parameters for eight loops from a refinery plant*

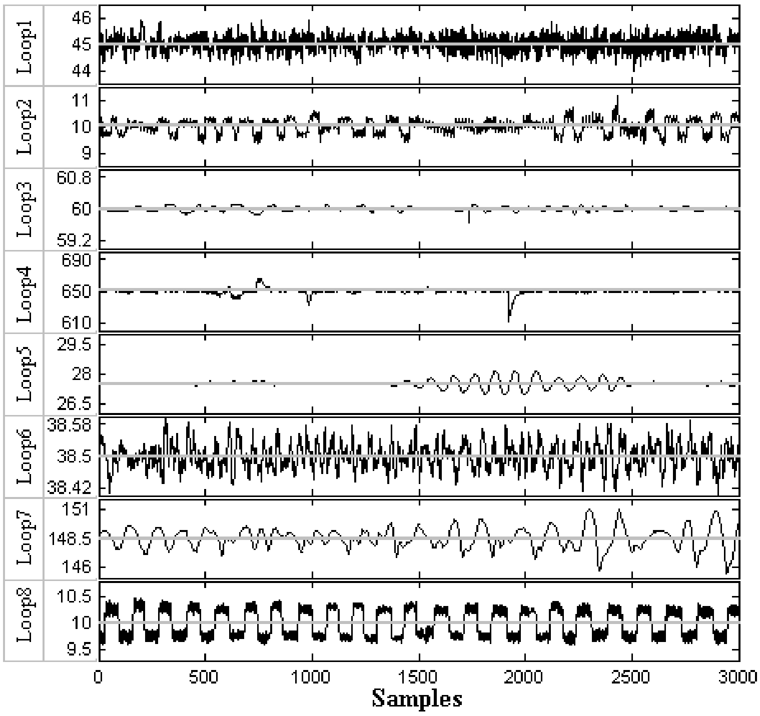


Figure 2.7: *Data sets of eight loops from a refinery plant: PV trends in black; SP trends in gray*

Modified Hägglund technique was applied trying different values for A_{Sin}, namely 1%, 2% and 5% of the control range; results are reported in table 2.2

Some loops (2, 5, and 8) show a clear oscillation, being positive to Hägglund technique application in the three cases ($\surd$ for each value of A_{Sin}); on the contrary loops 3, 4 and 6 give a negative response.

It should be noted that loop 6 has an oscillating behavior: it can be detected by applying autocorrelation function ACF [14], being able to eliminate the effect of

Loop	1	2	3	4	5	6	7	8
$A_{Sin} = 1\%$	$\surd$	$\surd$	X	X	$\surd$	X	$\surd$	$\surd$
$A_{Sin} = 2\%$	X	$\surd$	X	X	$\surd$	X	$\surd$	$\surd$
$A_{Sin} = 5\%$	X	$\surd$	X	X	$\surd$	X	X	$\surd$

Table 2.2: *Results of modified Hägglund Technique: $\surd$ oscillating, X not oscillating*

noise, enhancing the background dynamic of the loop; however not indicating its significativeness. In particular, ACF of loop 6 is characterized by values out of boundaries (tick gray lines in figure 2.8), meaning that on oscillatory behavior is present, once purged by noise.

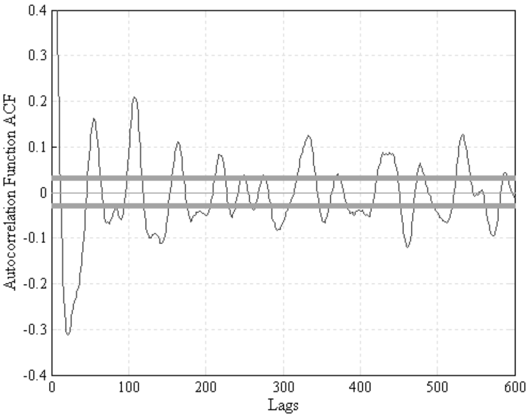


Figure 2.8: *Autocorrelation Function of loop 6 with boundaries (in thick gray lines)*

Modified Hägglund technique fails in recognizing the oscillation because of its amplitude, which has a maximum error of 0.08 units over a control range of 50 units: in this case the perturbation is correctly indicated as not significant.

For loops 1 and 7 results vary with A_{Sin}: with low values the oscillation is detected, for higher values the level of significance is larger and only few peaks in IAE_{Mod} trends overcomes the correspondent IAE_{Lim} (figure 2.9), thus labelling the loop as not oscillating.

Summarizing, modified Hägglund technique is able to detect oscillations rapidly for large amount of data; its advantage with respect to autocorrelation function is the classification based on the perturbation significativeness. The drawback due to the need of knowing the ultimate frequency of the system is partially eliminated by the application of a time moving window; however, it is suggested to adapt the threshold value IAE_{Lim} on the basis of operator knowledge whenever ω_U remains unknown.

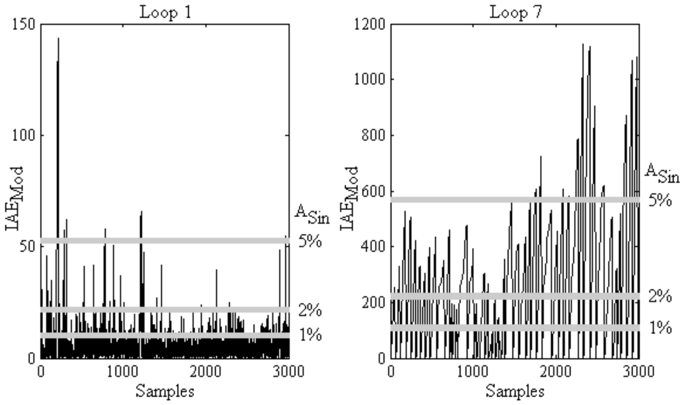


Figure 2.9: *IAE_{Mod} trends for loop 1 and 7 in black and IAE_{Lim} boundaries for different values of A_{Sin} in gray*

The output of this analysis is a classification of loops in two categories: *oscillating*, which will require a deeper analysis both in time domain and frequency domain (section 2.3), and *not oscillating*, which can be affected either by insulated anomalies (sluggish rejection of disturbances, section 2.2) or corrupted only by noise.

2.2 Identification of sluggish loops

In previous section a method used to detect oscillating loops is described; remaining loops can however be affected by other phenomena, as well as sluggish rejection of disturbances due to wrong tuning of controllers. The simple algorithm, which can be used to label those loops, is described in this section.

The basic idea is an extension of Hägglund technique [1]; again IAE_{Lim} is considered as a threshold value and IAE_{Mod} is evaluated. When the controller acts slowly a large peak in IAE_{Mod} is shown; therefore a loop is labelled sluggish when:

$$\max_t\{IAE_{Mod}(t)\} \geqslant IAE_{Lim} \cdot K_{SL} \qquad \text{(Eq. 3-8)}$$

where K_{SL} is assumed equal to 10 to maintain a certain symmetry with the oscillation detection technique, based on a limit value of $N_{Osc} = 10$.

Because the idea derived by Hägglund technique, its implementation can be based on the algorithm described in figure 2.5; its extension is shown in figure 2.10 with gray blocks. This modification is enough to describe entirely the problem; because of the algorithm structure, cycle 2 must be performed to apply Eq. 3-8: being

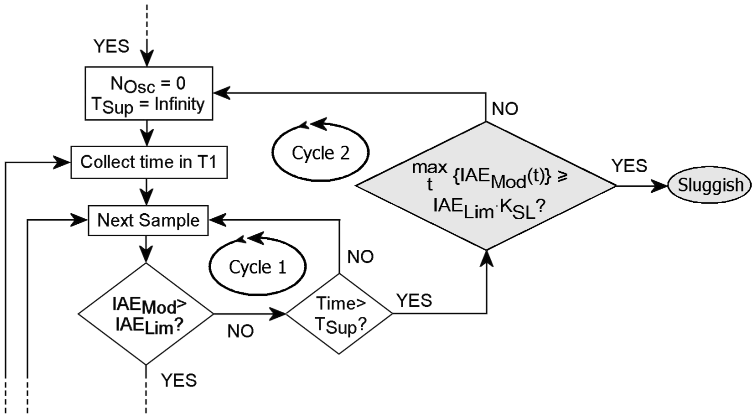


Figure 2.10: *Algorithm used to detect oscillations and its extension for sluggish loops (gray blocks)*

the whole procedure initialized with a value $T_{Sup} = \infty$, until a significant oscillation is not detected, actual time value will be surely lower than T_{Sup}, preventing the actuation of cycle 2. Henceforward two conditions must be satisfied:

1. an anomaly ($IAE_{Mod} > IAE_{Lim}$) must be detected, thus allowing the system to update the value of T_{Sup} with a finite real value; the algorithm has the possibility to exit from cycle 1 whenever no other anomalies are detected in the defined time moving window;

2. the exiting of the algorithm from cycle 1 to cycle 2, implies that no oscillating behavior is shown ($N_{Osc} < 10$), therefore the detected anomaly could only due to a disturbance rejection: if Eq. 3-8 is satisfied the loop is labelled sluggish.

Applying the algorithm depicted in figure 2.10 to same loops from refinery plant shown in figure 2.7, controller used in loop 4 is tagged as sluggish. Error trend is shown in figure 2.11a), having zoomed on abscissas to put in evidence the disturbance rejection response; correspondent IAE_{Mod} is represented in figure 2.11b).

With $A_{Sin} = 1\%$ and $IAE_{Lim} = 458.6$, maximum peak of IAE_{Mod} overcomes $K_{SL} \cdot IAE_{Lim}$ (dot gray line in figure 2.11b), thus satisfying Eq. 3-8.

The output of this analysis, including results from the algorithm depicted in figure 2.10, is represented by a classification of loops as: *oscillating*, as described in previous section, *sluggish*, to be counteracted with a new controller tuning, and *with good performance*, for those loops with negative response to the two tests, being corrupted only by noise.

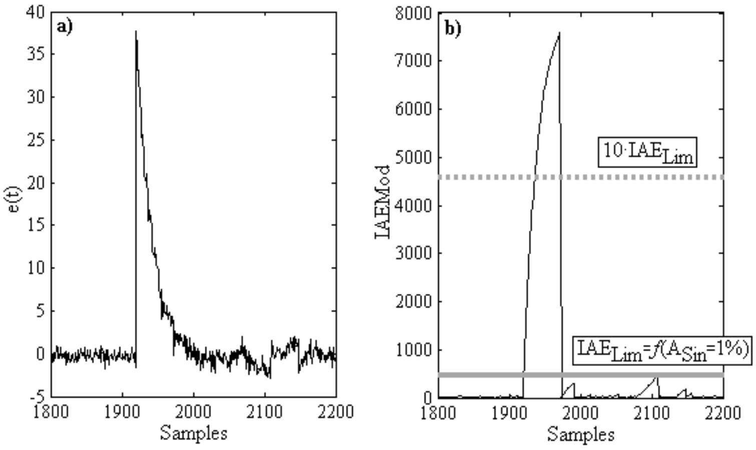


Figure 2.11: *Examples of sluggish controller: a) error e(t) trend; b) IAE_{Mod} trend and boundaries (IAE_{Lim} correspondent to $A_{Sin} = 1\%$ gray line, $10 \cdot IAE_{Lim}$ dot gray line)*

2.3 Classification of oscillating loops

In section 2.1 a technique to detect oscillating loops was described; on those loops a further analysis must be performed to indicate the cause of poor performance, choosing among sinusoidal disturbances, static friction on actuators, aggressive controller, marginal stability conditions etc. Both a time domain analysis and a frequency analysis are implemented: the former to study amplitudes of oscillation cycles, the latter to examine the energy of the signal as a function of all frequency channels. The overall algorithm, depicted in figure 2.12, will be described in the following subsections.

2.3.1 Damping test

The damping test is applied on all oscillating loops by performing three steps: to select the damped oscillation over the whole data set, to search its relative maxima and to calculate the best fit line passing through them.

As example, data depicted in figure 2.13 are used to put into evidence the efficacy of the damping analysis: error trends is shown, but only useful data are selected by applying an automatic procedure (white zone).

Boundaries of the white zone can be easily calculated: T_S is the first crossing zero before the highest peak in IAE_{Mod} trend, T_F is the crossing zero of the last significant oscillation detected in the time moving window, obtained by applying the algorithm in figure 2.10. To start from the highest peak in IAE_{Mod} has the

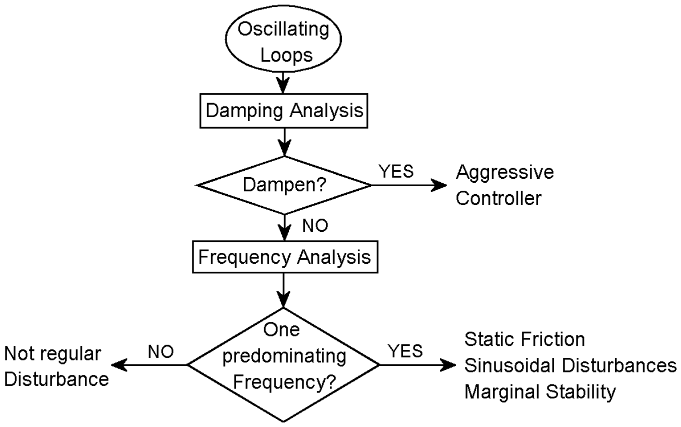


Figure 2.12: *Algorithm implemented to classify oscillating loops, using damping and frequency analysis*

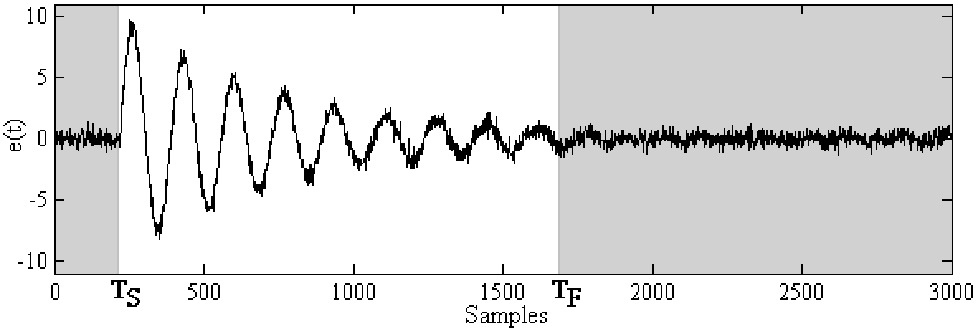


Figure 2.13: *Example of a damped oscillation, selection of data used in next analysis in white zone*

advantage of studying the anomaly in collected data with the largest ratio between disturbance and noise amplitudes, facilitating the following analysis. The drawback resides in the difficulty of extending this procedure to on-line cases, being built for an off-line application.

After data selection, the procedure calculates automatically relative maxima by studying each half-cycle of the damped oscillation and stores their values in an array R_{Max}. However, when the amplitude of the half-cycle reach the noise level, its relative maximum can assume an incorrect value. To avoid this problem Eq. 3-9 is assessed for the generic i^{th} half-cycle:

$$R^{Out}_{Max}(k) = \begin{cases} 1 & if \quad R_{Max}(k) \geqslant R_{Max}(i) \\ 0 & if \quad R_{Max}(k) < R_{Max}(i) \end{cases} \quad for \quad k = i+1..N \qquad \text{(Eq. 3-9)}$$
$$s.t. \quad k > 5$$

where N is the number of values stored in R_{Max}; R^{Out}_{Max} array is filled entirely by zeros when all half-cycles maxima from $i+1$ to N have values lower than the maximum of i^{th} half-cycle, thus indicating that the damped trend is still recognizable over the noise. On the contrary, if one point in R^{Out}_{Max} assumes unitary value (meaning that the correspondent half-cycle has a larger amplitude than i^{th} half-cycle) the damped trend is no more present: noise is starting to affect amplitudes of half-cycles. A minimum number of half-cycle maxima equal to five ($k > 5$) is requested to assess a proper best fit analysis (as shown in the sequel).

As consequence of Eq. 3-9, the last half-cycle to be considered for the damping test is $\#m$, calculated through Eq. 3-10:

$$\#m = \min_{k}\{[R^{Out}_{Max}(k) > 0] - 1\} \qquad \text{(Eq. 3-10)}$$

Relative maxima in the examined case are shown in gray circles in figure 2.14 a); last two half-cycles do not show the gray circles because noise almost covers the oscillation dynamic, thus not satisfying Eq. 3-9 and Eq. 3-10.

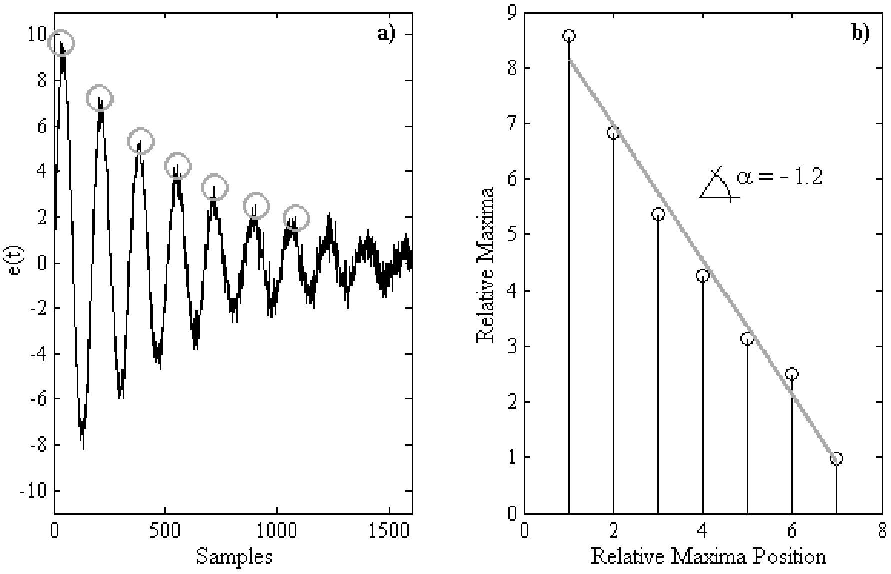


Figure 2.14: *Analysis of a damped oscillation: a) relative maxima detection (gray circles), b) best fit line*

Maxima are analyzed by applying a linear least square algorithm to obtain the best fit line represented in figure 2.14 b); if absolute value of its angular coefficient α is larger than a threshold value of $1/3$ the oscillation is considered damped.

2.3.2 Frequency test

Frequency test is applied on oscillating loops which are not labelled as damped; the main purpose is to study the energy of the signal at various frequencies shown by the *Power Spectrum* of the oscillation.

This analysis is based on discrete fourier transform FFT of the original signal; considering the error trend $e(t)$, its FFT is defined as:

$$FFT(k) = \sum_{j=1}^{N} e(j) \cdot \phi_R^{(j-1)(k-1)} \qquad \text{(Eq. 3-11)}$$

where ϕ_R is the R^{th} root of unity (Eq. 3-12) and N is the number collected samples.

$$\phi_N = e^{(-2\cdot\pi i)/N} \qquad \text{(Eq. 3-12)}$$

Power spectrum PS can be easily evaluated (Eq. 3-13), indicating with $conj(x)$ the complex conjugate of x; it is represented by a sequence of peaks: the larger their amplitude, the higher the influence of the correspondent frequency on the signal.

$$PS = \frac{FFT \cdot conj(FFT)}{N} \qquad \text{(Eq. 3-13)}$$

To explain the applicability of power spectrum in frequency studies, a sum of two sinusoidal function $A = sin(0.1 \cdot t) + 0.8sin(0.2 \cdot t) + RN$ will be considered, where RN indicates a random noise. In figure 2.15 the function (with RN increasing amplitude) and correspondent power spectra are shown.

To be noted that: 1) the two frequencies $\omega_1 = 0.1$ and $\omega_2 = 0.2$ correspond to power spectra major peaks (figure 2.15); 2) noise does not affect the result, being characterized only by high frequency channels with randomly distributed power.

Once power spectrum is calculated, major peaks must be distinguished from noise peaks. The algorithm, firstly used in [15], was developed at University of Alberta (Edmonton), in collaboration with Tangirala A.K., Dept. of Chemical Engineering, IIT Madras, Chennai, India (figure 2.16).

The algorithm starts with a filtering action, which eliminates the first three points of the power spectrum to avoid the effect of the signal mean, corresponding to $PS(\omega = 0)$. Having done that, a normalization is performed to obtain a normalized power spectrum NPS with unitary power:

$$NPS(k) = \frac{PS(k)}{\sum_{k=1}^{N} PS(k)} \qquad \text{(Eq. 3-14)}$$

NPS is sorted with increasing values and the flatness of its first derivative is evaluated by calculating the number of points N_{LD}, to whom corresponds a low

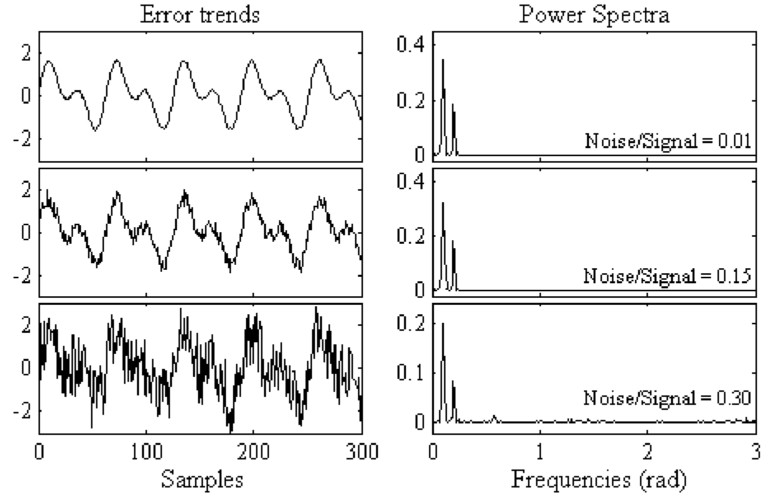


Figure 2.15: *Power Spectrum analysis on a sinusoidal function corrupted by random noise with increasing amplitude*

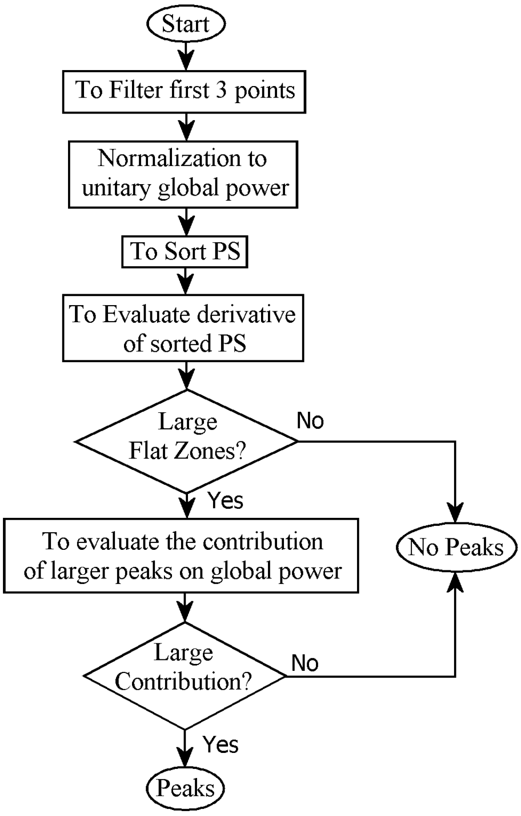

Start
To Filter first 3 points
Normalization to unitary global power
To Sort PS
To Evaluate derivative of sorted PS
Large Flat Zones?
No
Yes
To evaluate the contribution of larger peaks on global power
No Peaks
Large Contribution?
No
Yes
Peaks

Figure 2.16: *Algorithm used to detect major peaks in power spectrum trend*

derivative value:

$$N_{LD} : \left| \frac{dNPS_{Sorted}(\omega)}{d\omega} < 0.005 \right| \qquad \text{(Eq. 3-15)}$$

A large N_{LD} value indicates the presence of a wide range of frequencies associated to flat zones of derivative; because $\sum_{k=1}^{N} NPS(k) = 1$ for the normalization effect, only few frequencies give a major contribution to the power spectrum: for this case there is the possibility of having sharp peaks. On the contrary, for low N_{LD} values (few flat zones), the contribution to the unity is spread over a larger number of peaks with similar power, thus excluding the presence of a predominating sharp peak. By default, a tolerance of flatness of 90% is chosen (meaning that $N_{LD}/N > 90\%$ indicates large flat zones).

A further test must be performed to evaluate the contribution of peaks not included in flat zones; when low, the presence of the peak is insufficient to label its frequency as predominating. Indicating with $A_P(m)$ the sum of powers of the first larger m peaks (by default $m = 4$), the contribution to unity power is expressed as:

$$C(m) = \frac{\sum_{k=1}^{N} NPS(k)}{A_P(m)} = \frac{1}{A_P(m)} \qquad \text{(Eq. 3-16)}$$

When $C(m)$ overcomes 1.5, meaning that the first larger m peaks contribute more than 50% to the unity power, the presence of sharp predominating peaks is indicated. The use of three parameters as well as threshold on derivative values, percentage of flatness and number of major peaks to be detected, makes the algorithm very flexible. Default values here reported (0.005, 90% and 4 respectively), allows the system to obtain a good compromise between discharging loops oscillating at a well defined frequency and analyzing oscillations with mixed frequencies.

2.3.3 Application of Frequency and Time domain tests

In this section an application of the two described tests is shown, considering both plant data (figure 2.17) and simulated data (figure 2.18).

Globally eight loops are shown: loops 1 and 6 are affected by noise; loops 2, 3, 4 and 7 show an oscillatory behavior; loop 5 and 8 simulate respectively a sluggish controller and an aggressive controller. Results are summarized in table 2.3: time domain analysis reveals correctly a damped oscillation only in loop 8, labelling not damped all other loops. Frequency test is positive for loops 2, 3, 4, 7 and 8, whose power spectra show a sharp peak; loops 1 and 6 have a sequence of peak at different frequency, but with similar amplitude; loop 5 has a peak at low frequencies but its smoothness exclude an oscillatory behavior.

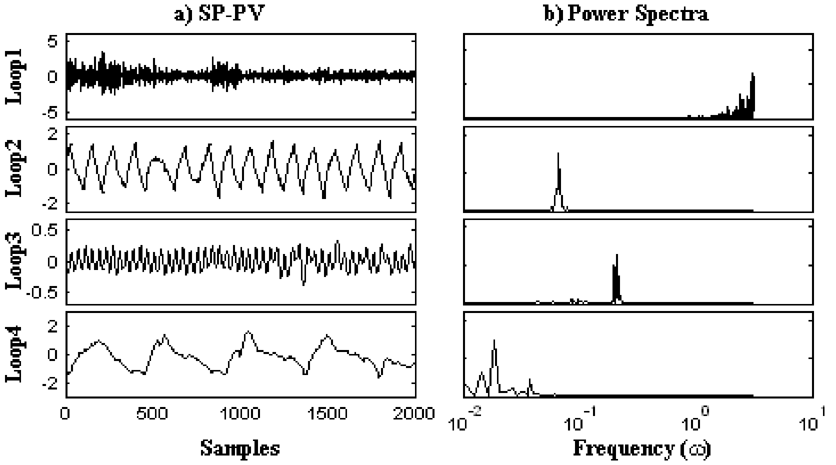


Figure 2.17: Plant data: a) error plots, b) power spectra

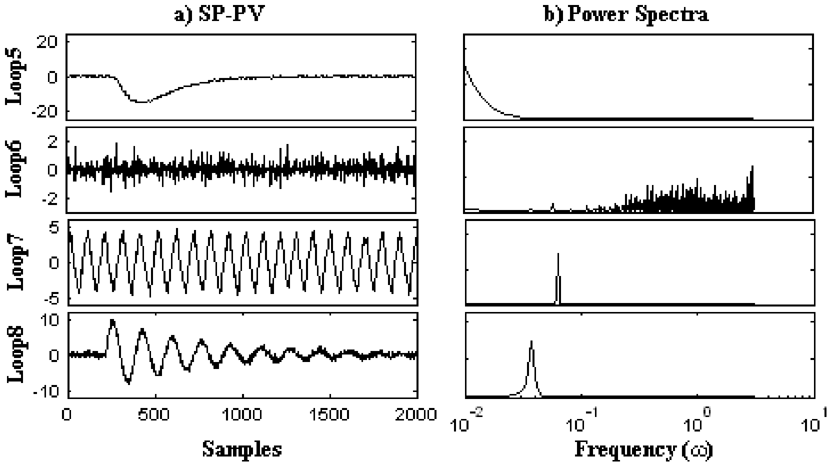


Figure 2.18: Simulated data: a) error plots, b) power spectra

Table 2.3: Results from the application of frequency and time domain tests

Loop	1	2	3	4	5	6	7	8
Damped	No	No	No	No	No	No	No	Yes
Freq.Peak	No	Yes	Yes	Yes	No	No	Yes	Yes

Summarizing, two test can be performed in sequence to detect damped oscillations and to evaluate the frequency content of the signal, thus putting into evidence loops with a predominating frequency. Consequently, the output of this analysis is a classification among oscillating loops: *Damped oscillation*, to be counteracted by tuning the controller (see chapter 4); *Oscillation without predominating frequency*, due to external disturbances to be eliminated by plant analysis; *Oscillation with a predominating frequency*, to be analyzed more in detail as described in chapter 3.

Chapter 3

Stiction on Actuators

The previous chapter describe some techniques to detect loops affected by anomalies, namely significant differences between controlled variable PV and set-point value SP, distinguishing them from loops affected only by noise. Hence a first classification is done, tagging loops with *good performance*, *oscillating* and *sluggish controller* labels. Further tests are applied over oscillating loops, both in frequency and time domain, to obtain the classification amongst: *damped oscillation*, oscillation with *one predominating frequency*, oscillation with *no predominating frequency*. In case of not damped oscillation, well localized in frequency, the cause of poor performance can be searched among marginal stability conditions, external sinusoidal disturbances, or static friction on valves. Neglecting the first item, however detectable by a direct comparison between oscillating frequency and ultimate frequency ω_U, it becomes fundamental to distinguish between the two last phenomena. Particulary, static friction (*stiction*) on actuators is a very common problem on industrial plant [6]. This fact can be counteracted by performing a valve maintenance, more frequent action, or by adopting friction compensators [17], [18]. As explained before, set point SP, controlled variable PV, and controller output OP are generally recorded by DCS, while manipulated variable MV is not. This is a key assumption of this chapter, which has implication in the choice of the techniques described in next sections to be used for automatic detection of stiction.

3.1 Stiction analysis

The presence of stiction in actuators introduces both a delay and a non-linear behavior between the control signal OP and manipulated variable MV, as qualitatively illustrated in the sequel.

Static friction effect has been studied through the balance of forces around the

[*] Portions of this chapter are published in [8], [9] and [16]

actuator: simplifying the problem, when the active force F_A due to the controller output is less than the static friction force F_S the valve is completely stuck. This can cause an offset between process variable PV and set-point value SP; the constant error, in presence of an integral component in the controller, brings to an increase of OP and, as consequence, to the active force. When F_A overcomes F_S the valve suddenly moves, jumping to a new position. Changes in MV values, at this point affected only by dynamic friction, may become larger than necessary, thus causing a change in error $SP - PV$ sign. The controller decreases its action and the active force becomes again less than the static friction force: the valve gets stuck in a new position.

The sequence motion/stop of the valve due to this phenomena is called *stick/slip motion* and brings to a characteristic trend in MV. In case of predominance of static friction force, hence having a stuck valve, MV is constant with time even though OP increases because of the controller action. Energy, stored in actuator during this phase, is suddenly released when active force becomes stronger than stiction and MV jumps almost immediately to a new value. At this point the motion phase starts and MV varies gradually; active force is only counterbalanced by dynamic friction force.

MV trends are depicted in figure 3.1a) as function of OP values and 3.1b) as function of time.

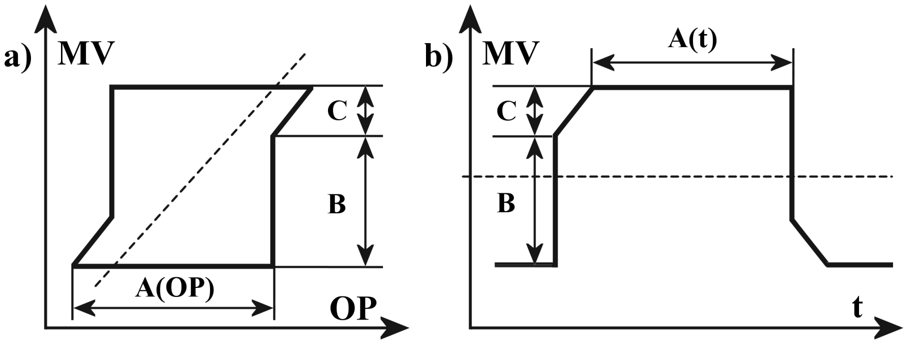


Figure 3.1: *a) Characteristic curve of the valve; b) correspondent trend of $MV(t)$. Case of Stiction (solid line), desired performance (dashed line)*

Manipulated variable shows an oscillating trend, that can be almost approximated to a square-wave shape, when neglected the short period of the motion phase (**C** in figure 3.1). This trend is also shown in PV, but filtered by the process dynamic. Oscillations on PV are the cause of low control loop performance.

3.2 Stiction models

Modelling stiction is one of most important issues of this chapter; the importance of generating a model resides in the possibility of a detailed study of this phenomena. Models should have some main characteristics: simplicity, to be easy applicable and understandable and reliability, to give a better interpretation of main dynamics of phenomena.

Many models have been proposed in literature to describe the presence of friction in actuators; a survey is reported in [19]. Two typical approaches can be chosen:

- **physical models**: which try to extrapolate fundamental laws to describe stiction through the application of a balance among all forces
- **data-driven models**: which try to simulate the presence of stiction by reproducing trends (figure 3.1a) using formulas to translate OP values in correspondent MV values.

One of universally accepted physical models is the Karnopp model [20]; on the contrary among data-driven models the Choudhury model [21] has been chosen for its simplicity.

3.2.1 Karnopp model

Even though this model is based on a balance of forces, for its nature very complex, the application is relatively simple as shown by following equations:

$$M \cdot \frac{d^2x}{dt^2} = F_A - F - F_K \qquad \text{(Eq. 4-1)}$$

$$F_K = k \cdot x \qquad \text{(Eq. 4-2)}$$

$$F = \begin{cases} -F_D sign(v) + F_V v & if \quad v < -\Delta V \\ max[-F_S, F_A] & if \quad -\Delta V < v < 0 \\ min[-F_S, F_A] & if \quad 0 > v > \Delta V \\ F_D sign(v) + F_V v & if \quad v > \Delta V \end{cases} \qquad \text{(Eq. 4-3)}$$

Eq. 4-1 represents the balance between the mass of the actuator M, multiplied by an acceleration d^2x/dt^2, and three forces: active force F_A, friction force F and opponent spring force F_K. The first one, as explained in the previous section, is generated by the controller through a signal 4-20 mA, which is translated in a pressure on valve and consequently a force over the actuator. Spring force is proportional to the position of the actuator x (Eq. 4-2), while friction force has four contributions depending symmetrically on speed of the valve $v = dx/dt$ (Eq. 4-3) and a threshold value ΔV. For speed values between $|\Delta V|$ and zero, the active force is perfectly balanced by counteracting forces, thus obtaining a stuck valve. Out of this range, only dynamic friction force is present, as function of speed v.

The active force, counterbalanced by the other contributions F_K and F, generates a cycle with the actuator position in case of stiction: firstly the valve starts to move (x increasing or decreasing) and force on actuator also varies; when the error, and consequently active force, changes its sign, speed decreases reaching the threshold value ΔV; at this point static friction becomes preponderant and the valve is stuck (x constant, force on actuator varying). An example of this cycle is depicted in figure 3.2.

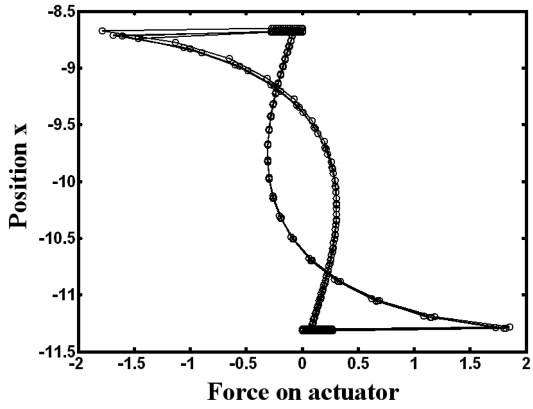


Figure 3.2: *Case of a valve affected by stiction, force on actuator as a function of its position, Karnopp model*

The forces balance can be properly represented by the Simulink scheme depicted in figure 3.3.

To simulate action of forces over the actuator, a first derivative block $1/(M \cdot s)$ is implemented to obtain the speed signal v, to be used for dynamic and static friction evaluation. A second derivative block $(1/s)$ transforms speed in position x, input of the spring force F_K. Between speed and forces summation point two blocks are used: *Dead zone* block, whose parameters *start of dead zone* and *end of dead zone* are chosen equal to $-\Delta V$ and $+\Delta V$ respectively; *Columbus & Viscous Friction* block, whose parameters *offset* and *gain* are chosen equal to ΔV, and 1 respectively. The effect is a speed $v = 0$ when input signal lies in $[-\Delta V;+\Delta V]$ and a jump to $\pm\Delta V$ when the valve starts to move again.

Same sequence of blocks can be implemented to simulate dynamic friction; the main difference is in *Columbus & Viscous Friction* Block where offset and gain are respectively F_D and F_V (with reference to Eq. 4-2).

Static friction is more complex because of the introduction of several nonlinearities. The multiplying block receives as inputs the active force F_A and the

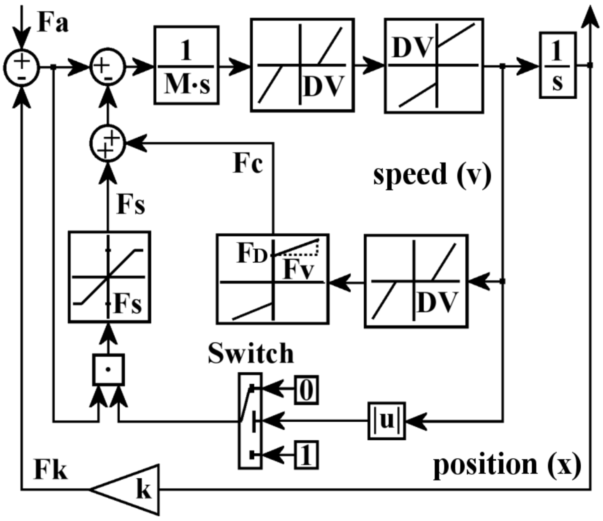


Figure 3.3: *Simulink scheme for Karnopp stiction model*

output of a *Switch* block. When absolute value of speed is under the threshold value ΔV, the switch block goes in position one, activating through the multiplying block the static friction force. On the contrary, for a valve in motion ($|v| \geqslant \Delta V$), the switch is in position zero and static friction is no more present. The effective contribution of static friction force is represented by the output of a *Saturation* block, limited to [$-F_S$,$+F_S$] values.

Even though Karnopp simulation scheme is the simplest among models described in [19], a relatively large number of parameters must be managed: M mass of actuator; F_S static friction force value; F_D, F_V to define completely dynamic friction force; ΔV speed threshold value. Furthermore some difficulties arise in choosing proper values for some parameters as well as the mass of actuator, which depends on the typology of used valves, or dynamic friction force, which depends on processed products.

3.2.2 Choudhury model

Previous described model has a drawback in the number of parameters; a different approach has been adopted by Choudhury et al. [21] with a *data-driven* model. Basic idea is to reproduce through the application of a mathematical function the characteristic curve of a valve affected by stiction. Considering typical trends in MV, as depicted in figure 3.1, only two parameters can be used to describe this phenomenon: S, a direct measure of stiction, and J, value of the jump done by the valve when starts to move. Varying values of S and J different cases can be simulated; namely *undershooting* stiction (figure 3.4a) and *overshooting* stiction (figure 3.4b). In the first case the valve, after jump, does not reach the position it should have in absence of stiction ($J < DP$). On the contrary, in the second case, the energy accumulated in the valve when stuck causes a larger jump ($J > DP$), thus generating a more complex cycle to be analyzed.

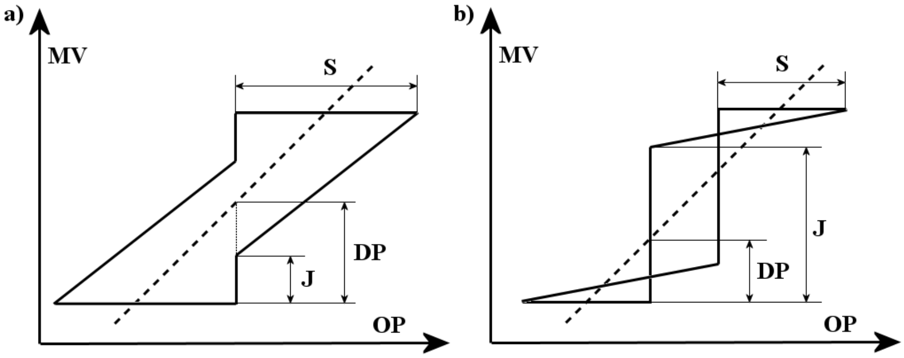


Figure 3.4: *Different cases of stiction modelled by Choudhury: a) undershooting; b) overshooting*

The drawback of loosing physical correspondence to summation of real forces, as Karnopp model uses, is largely counterbalanced by Choudhury model simplicity: S and J parameters can describe perfectly main stiction dynamics.

Simulink scheme of the valve is depicted in figure 3.5.

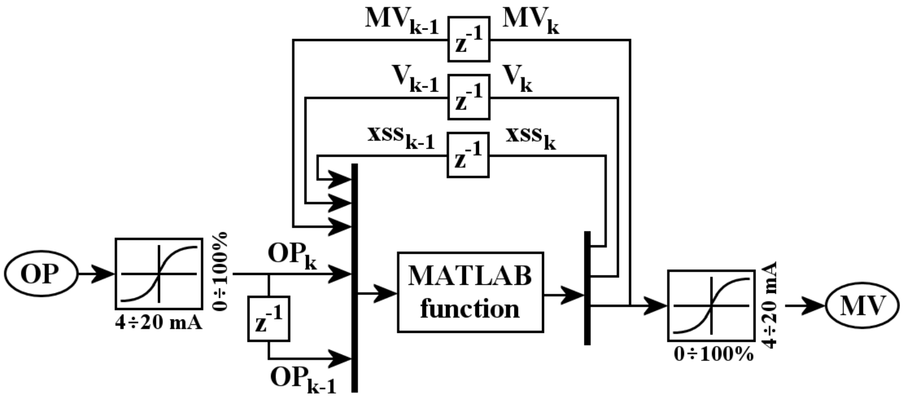


Figure 3.5: *Choudhury model: Simulink scheme of the valve*

The scheme is composed by two *look-up table* blocks to convert a $4\,mA \div 20\,mA$ signal to a $0\% \div 100\%$ signal and vice versa. The core of the scheme is represented by the Matlab function, whose inputs are: controller output and its previous sample (OP_K and OP_{K-1} respectively); last recorded values of manipulated variable (MV_{K-1}); last recorded speed of actuator (V_{K-1}) and a variable (XSS_{K-1}) used in the algorithm as shown in the sequel. Outputs are represented by new calculated values of MV, XSS and V. To delay signals by one sample a *Unit delay* block (Z^{-1}) is used. The algorithm used in *Matlab function* block, is described in detail in figure 3.6.

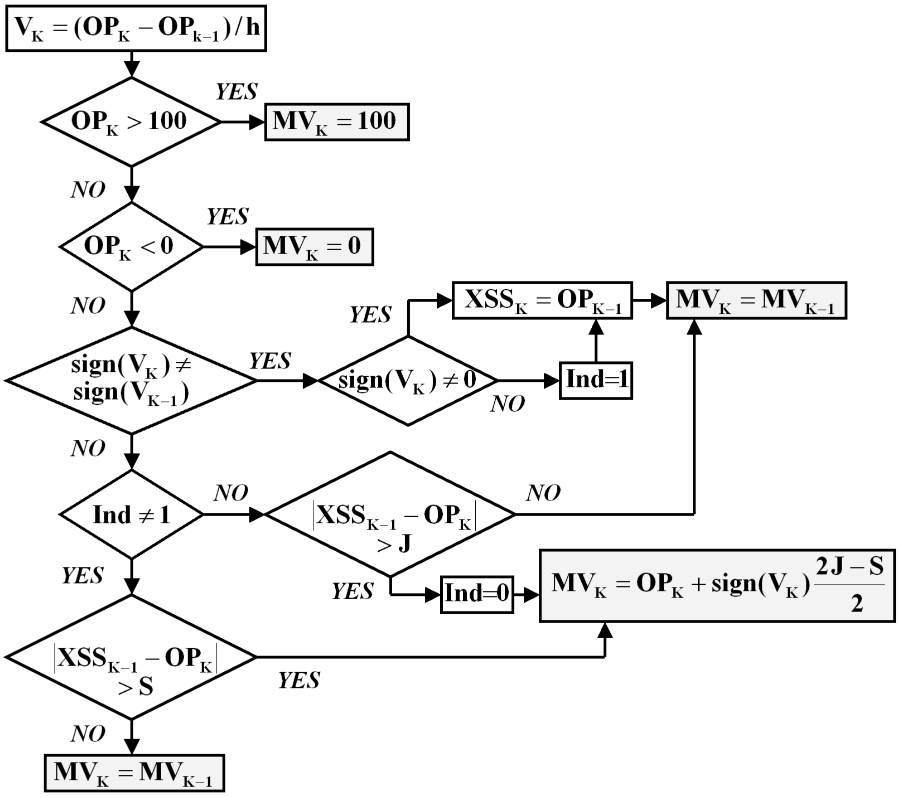


Figure 3.6: *Choudhury model: algorithm of the Matlab function block*

The first block evaluates the new valve speed, as variation in OP value divided by the sampling time h. Following question blocks generates saturation on valve when input signal OP goes out of the range $[0\% \div 100\%]$; associating to MV 0% and 100% values respectively. A closing or opening valve ($sign(V_K) = sing(V_{K-1})$ equal respectively to -1 and +1) is related to a situation of not stuck actuator ($|XSS_{K-1} - OP_K| > S$), or actuator in jump phase ($|XSS_{K-1} - OP_K| > J$). Variable Ind is used internally in the proposed algorithm: unitary value comes from a change in speed sign:

$$(OP_K - OPk - 1) \neq (OP_{K-1} - OP_{K-2}) \Rightarrow Ind = 1 \qquad \text{(Eq. 4-4)}$$

Int is set again to zero after the jump ($|XSS_{k-1} - OPk| > J$), when the valve starts to move (equation Eq. 4-5. The use of the internal variable Ind is due to

Simulink approximations, whose effect is an error on OP signal of magnitude order of machine precision: to counterbalance the generated fictitious speed, the valve is considered stuck ($MV_K = MV_{K-1}$).

$$MV_K = OP_K + sign(V_K) \cdot \frac{2J - S}{2} \qquad \text{(Eq. 4-5)}$$

3.2.3 Stiction simulation

The described models were used to simulate stiction in order to understand its effect on recorded variable (OP, MV, PV) and to design an automatic detection technique.

As first step, a comparison between models was made: in this section the case of a first order plus time delay ($FOPTD$) model will be considered; more extended analysis will be reported in the sequel. Reference process model and adopted Ziegler-Nichols [13] controller are:

$$P = \frac{1 \cdot e^{-10s}}{10 \cdot s + 1}; \quad C = 0.92 \cdot \frac{25 \cdot s + 1}{25 \cdot s} \qquad \text{(Eq. 4-6)}$$

As starting point, Choudhury model parameters were chosen equal to $S = 5$ and $J = 4$. To those values correspond a Karnopp model with unitary mass actuator ($M = 1$) and a threshold value $\Delta V = 0.2$; forces were defined as: static friction force $F_S = 1$; dynamic friction force with $F_D = F_V = 0.5$; spring force with $k = 0.7$. Graphical comparison of recorded values is shown in figure 3.7.

The amplitude of variables has no importance in this analysis; for instance PV amplitude in Choudhury model can be easily uniformed to Karnopp model by changing S value without affecting other characteristics of oscillations. OP and MV amplitudes depend only on scale factors used to pass from mA to percentage signals.

Neglecting amplitudes of signals, same trends in terms of shape can be recognized (figure 3.7), thus demonstrating the possibility to use both models indistinguishably. It must be considered also that changing one parameter in Karnopp model, as well as mass actuator or threshold value on speed, and adopting different values of friction and spring forces brings to same trends. This indetermination in parameters highlights the easy applicability of the Choudhury model with respect to the Karnopp model; only the former will be applied in the sequel.

Oscillation shape and frequency were studied as function of process parameters (time delay θ, time constant τ), stiction parameters (S, J) and controller parameters (principally gain K_C). Typical results are shown in figure 3.8, 3.9 and 3.10.

It can be pointed out from examined cases that characteristic wave shapes can be associated to stiction phenomena for the three variables OP, MV, PV. Particularly, manipulated variable MV is affected slightly by varying parameters and

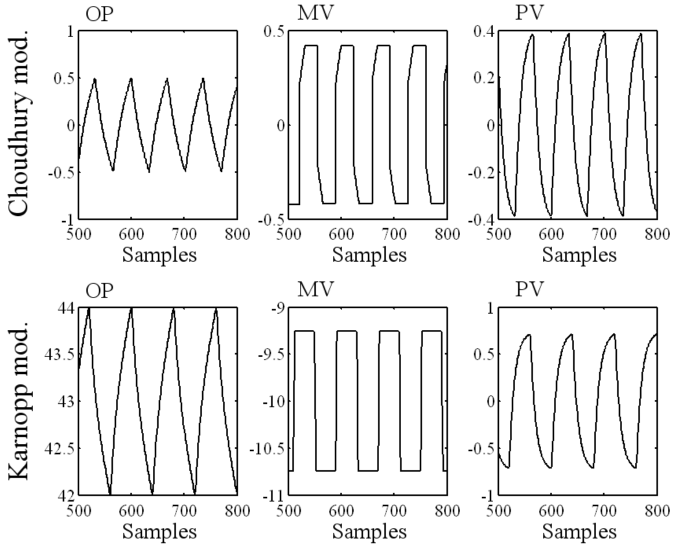


Figure 3.7: Comparison between Karnopp and Choudhury models with a FOPTD model; time constant and time delay equal to 10

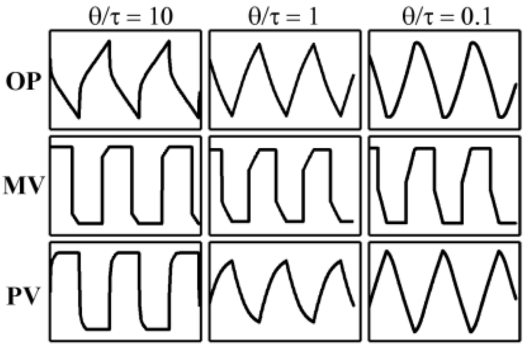


Figure 3.8: *OP, MV and PV trends as function of ratio θ/τ*

maintains in all cases an approximatively square wave. Also OP is generally almost unaffected and a triangular wave can be recognized in all trends. On the contrary, controlled variable PV assumes different shapes: from square waves, in case of time delay θ over time constant τ (figure 3.8) or high S/J ratio (figure 3.9), to triangular waves for opposite cases, passing through intermediate shapes. In case of square waves stiction is easy recognizable from sinusoidal disturbances, on the contrary, triangular waves, especially with noise, could be not distinguished.

In figure 3.10 controller gain K_C is changed to evaluate its impact over trends of collected variables: a variation in K_C has the peculiarity of affecting oscillation frequency, without influencing its shape: a decrease from K_C to $(2/3)K_C$ to $(1/2)K_C$

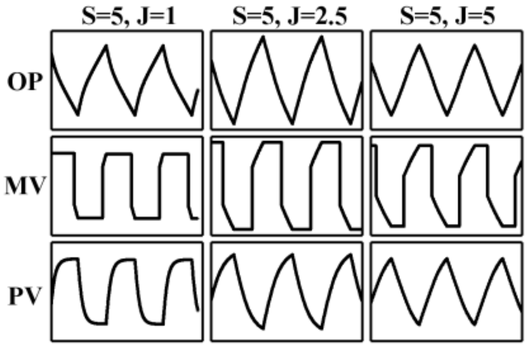


Figure 3.9: *OP, MV and PV trends as function of ratio S/J*

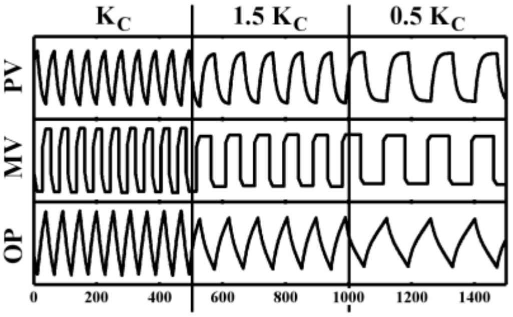


Figure 3.10: *OP, MV and PV trends as function of K_C*

causes a decrease in oscillation frequency from ω_0 to $0.58\omega_0$ to $0.38\omega_0$; indicating with ω_0 the starting frequency. An intuitive explanation of this phenomena resides on force balance used in Karnopp model: a decrease in K_C means a lower active force F_A because of a direct proportionality; therefore a longer time is requested to overcome stiction force F_S and consequently a lower frequency of oscillation is shown.

3.3 Stiction Techniques

In recent literature, several different techniques for detection of stiction have been proposed. Some of them are manual or require strong interaction with the operator; for instance [1]. Other techniques are based on additional knowledge about the characteristic curve of the valve and values of manipulated variable or valve position [22], [23]. Among techniques which require only the knowledge of OP and PV, with limited additional requirements for process and valve parameters (and therefore more suitable for automatic implementation in a closed loop performance monitoring system), some other contributions should be mentioned: [24] makes use of a model based non linear observer (EKF type), [25] proposes a very simple method based on a Cross-Correlation between OP and PV, while [26] considers the probability distribution of the PV derivatives, thus extending the previous method to integrating processes. [27] uses a neural network to approximate the process output by means of 7 different elementary shapes (primitives), then associating square and triangular waves to the presence of stiction. [28] focuses on discontinuities of MV derivatives (jumps of the valve); a segmentation-based method is used to evaluate the distribution of jumps as a function of their magnitude: larger magnitude indicates stiction. [29] proposes a technique based on the bicoherence function of PV: the presence of peaks is a symptom of stiction. [9] approximates the PV trend with relay-type, triangular and sinusoidal waves: the first two cases are associated to a stiction phenomenon.

A common characteristics of the last techniques is that they perform a fitting of PV data, in order to detect the typical signature of stiction and distinguish it from other causes.

The interest towards automatic techniques to be applied in a performance monitoring system, without any specific knowledge or additional requirements for data acquisition, forced to focus on *Cross-Correlation* ([25]), *Bicoherence* ([29]) and *Relay technique* ([9] and [16]). These techniques will be briefly illustrated and the analysis of their applicability will be performed in the sequel, with the goal of putting into evidence limitations coming from the lack of specific knowledge on valve and process characteristics.

3.3.1 Cross Correlation Technique

Horch ([25]) observed that the phase shift between the control output OP and the process variable PV depends on the presence of stiction. In particular with a proportional-integral controller: in presence of stiction, control signal OP and controlled variable PV oscillate with a phase shift $\phi = \pi/2$, in presence of a sinusoidal perturbation the phase shift is $\phi = \pi$. Using the cross correlation function between PV and OP two parameters $\Delta\tau$ and $\Delta\rho$ are estimated (see Appendix A). According to the values of these two parameters, conditions corresponding to the presence of stiction or sinusoidal perturbations can be indicated; also an uncertainty range, where no decision can be taken, is encountered (gray zone). Results are summarized in Table 3.1.

Table 3.1: *Indications from the Cross Correlation technique*

No Stiction	Uncertainty	Stiction
$0 < \Delta\rho \leq 0.072$	$0.072 < \Delta\rho \leq 1/3$	$1/3 < \Delta\rho \leq 1$
$0 < \Delta\tau \leq 1/3$	$0 < \Delta\tau \leq 2/3$	$2/3 < \Delta\tau \leq 1$

An example of cross correlation application is shown in figure 3.11 for a case of stiction and in figure 3.12 for a sinusoidal disturbance:

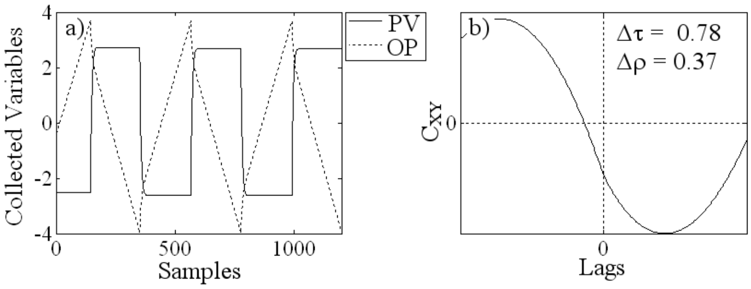


Figure 3.11: *Application of Cross Correlation technique in a case of stiction: a)trends of PV and OP; b) trend of cross-correlation*

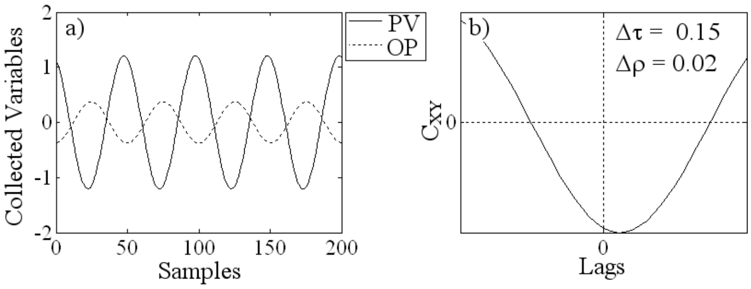


Figure 3.12: *Application of Cross Correlation technique for a sinusoidal disturbance: a)trends of PV and OP; b) trend of cross-correlation*

As case study, stiction is modelled using Choudhury [21] approach, setting $S = 7$ and $J = 1$ for a $FOPTD$ process with $\theta = \tau = 10$; sinusoidal disturbance is characterized by a unitary amplitude and frequency $\omega_D = 0.05 rad/sec$. Both cases are correctly indicated: $\Delta\tau = 0.78$ and $\Delta\rho = 0.37$ for stiction; $\Delta\tau = 0.15$ and $\Delta\rho = 0.02$ for sinusoidal disturbance. Those results will be extended in the sequel, using different values of (S, J), disturbance frequencies and θ/τ ratio.

An alternative approach [26] also considers the case of unstable (integrating) processes, by making use of the sampling histogram of the second derivative of PV. After building the histogram two approximating probability distributions are used: an approximatively gaussian distribution indicates stiction; a bimodal distribution indicates sinusoidal disturbance. Even though this technique is characterized by the big advantage of collecting only PV, it has a main drawback in the passage through two derivatives; henceforward noise affects heavily the procedure, being enhanced by the derivation process. To avoid a complete hiding of main wave shape by noise, some filters are used: their effect reduces the problem, but also changes original wave shape of PV, thus reducing efficiency of the method [30]. For this reason, and related difficulties in defining unique values of filter constants for all typology of processes and noise, this technique is not considered in present study.

3.3.2 Bicoherence Technique

This technique has been recently proposed and is based on High Order Statistic [29]. It is observed that the first and second order statistic (mean, variance, autocorrelation, power spectrum, etc.) are only sufficient to describe linear systems. Non linear behavior must be detected using higher statistics such as *Bispectrum* and *Bicoherence*. To be noted that this technique has been introduced to detect non-linearities, in general; application to stiction phenomenon can be considered as a particular case. A more detailed explanation of basic equations, including also *Trispectrum* extension, can be found in [31].

The bicoherence, based on the Fourier transform of the data, is a measure of interaction of frequencies: its plot shows large peaks for frequencies that are interacting indicating a strong non linearity. Two tests can be performed on this function: the first, based on an index named NGI, indicates when a signal is gaussian ($NGI < 0.001$); the second, based on an index called NLI, indicates when the signal is non-linear ($NLI > 0.1$). If the signal is gaussian or linear the control loop is not affected by stiction; (more details in Appendix A).

Application of the bicoherence technique on the case study, described in the previous section, is shown in figure 3.13 for stiction and in figure 3.14 for sinusoidal disturbance.

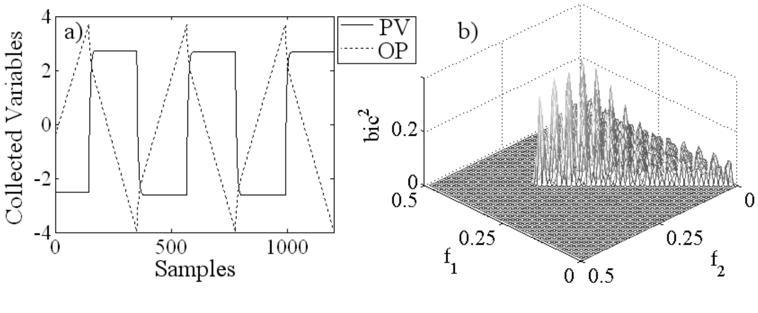


Figure 3.13: Application of Bicoherence technique in a case of stiction: a)trends of PV and OP; b) bicoherence function

Graphically, the presence of stiction is clearly demonstrated by the sequence of peaks in figure 3.13b), not shown in case of disturbance 3.14b). The two indexes confirm this result: in the first case $NGI = 0.067$ indicates a non gaussian signal

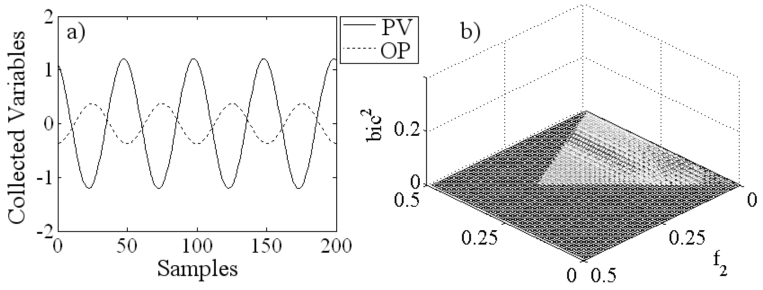


Figure 3.14: *Application of Bicoherence technique for a sinusoidal disturbance: a)trends of PV and OP; b) bicoherence function*

and $NLI = 0.28$ highlights the presence of a non linear behavior; in the second case $NGI = -0.15$ excludes automatically the presence of stiction, being the signal gaussian.

3.3.3 Relay Technique

In previous sections trends for OP, PV and MV were described and the typical square wave on MV and the different types of waves in PV were observed. In a slight different scenario (closed loop relay identification for *autotuning* purposes [32], [33]) wave shape similar to the ones reported in figures 3.8 and 3.9 have been observed for loop variables of a $FOPTD$ process. The shape of signals is modified by process dynamics: this suggests the basic idea of the technique proposed here. Every significant half cycle of the recorded oscillation is fitted by using three different models: the output response of a first order plus time delay under relay control, a triangular wave and a sine wave. By adopting a similar interpretation to Horch [25], relay and triangular waves are associated with the presence of stiction, while sinusoidal shape with the presence of external perturbations. By comparing the error between real and fitted data, an evaluation of the accuracy of approximation and then an indication of the underlying phenomenon observed in the analyzed variable can be obtained. More details for this procedure are given in Appendix A.

Once approximations have been performed, a stiction index (S_I) can be defined. Being E_S the minimum square error obtained by the sinusoidal approximation and E_{RT} the one obtained by the better approximation between the relay and the triangular waves, S_I can be defined as:

$$S_I \triangleq \frac{\overline{E_S} - \overline{E_{RT}}}{\overline{E_S} + \overline{E_{RT}}} \qquad \text{(Eq. 4-7)}$$

where the generic symbol $\overline{X}$ indicates the average value of x on all the examined half-cycles. S_I varies in the range [-1; +1]: negative values indicates a better approximation by means of sinusoids, positive values by means of relay or triangular approximations. Values close to zero indicate that the two approximations have similar errors and the procedure gives an uncertain answer; by considering that noise can change the shape of the curve, the uncertainty zone is defined by $|S_I| < 0.21$. This corresponds to the ratio $\overline{E_S}/\overline{E_{RT}} = 0.66$ [9], in analogy to the limit value $\Delta\tau \leq 2/3$ [25].

S_I can be applied separately either on positive half-cycles ($PV > SP$, thus obtaining an index S_I^+) or negative half-cycles ($PV < SP$, thus obtaining an index S_I^-). A direct comparison between values of S_I^+ and S_I^- helps in identifying asymmetric stiction, where opening and closing phases of the valve generate different wave shapes on positive and negative half cycles (nearer to sinusoids where the effect of stiction is weaker).

Applying this technique on the case study, approximating curves are shown for three half cycles in figure 3.15 for the case of stiction.

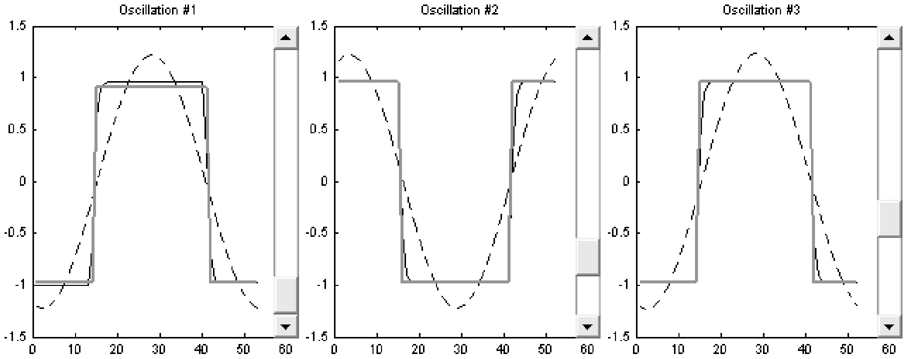


Figure 3.15: *Application of Relay technique in case of stiction. Approximating curves for three half-cycles: collected data (black thin), relay curve (gray thick), sinusoidal curve (black dots)*

Graphical trends, which highlight the better approximation of the square wave, are supported by indexes values $S : I^+ = 0.786$, $S_I^- = 0.782$ and $S_I = 0.784$; thus indicating clearly the presence of stiction.

For the case of sinusoidal disturbance, similar trends are shown in figure 3.16. In this case, sinusoidal approximation and collected data are almost overlapping; stiction indexes $S_I = S_I^+ = S_I^- = -1$ indicate absence of stiction.

3.4 Comparison of different techniques

Cross Correlation technique was firstly considered because of its simplicity and fast applicability on data sets, with a direct application on plant data furnished by ENI R&M - Refinery of Livorno, Italy. Examined data refer to a flow control loop, with

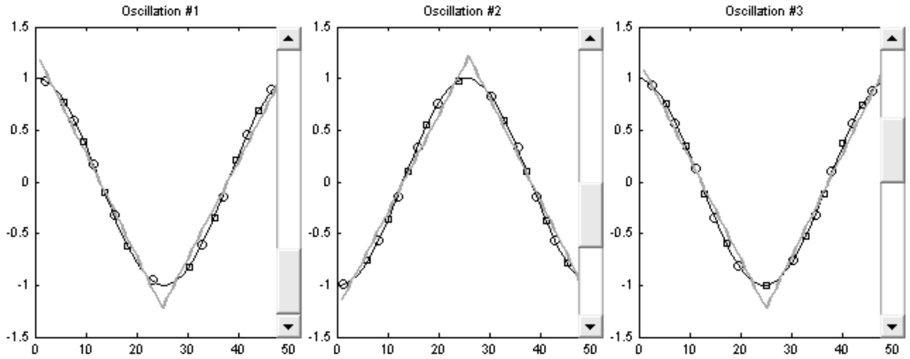


Figure 3.16: *Application of Relay technique for sinusoidal disturbance. Approximating curves for three half-cycles: collected data (black thin), relay curve (gray thick), sinusoidal curve (black dots)*

evident presence of valve stiction and oscillation in controlled output PV. In figure 3.17 and 3.18 sets of recorded data in two different time windows are shown; in the upper part of both figures PV (thick line) and OP (thin line) are reported; in the lower part cross correlation trend (dash line) between OP and PV is presented.

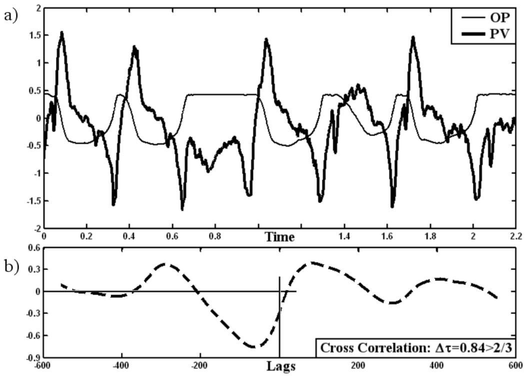


Figure 3.17: *Application of Cross correlation on data set* A*: a) PV and OP trends, b) Cross Correlation trend*

In the examined control loop, Horch technique was applied choosing a variable number of oscillation with different time windows. The number of analyzed oscillations seems not to influence the result and in most cases stiction is correctly detected ($\Delta\tau = 0.7 \div 0.85$; as for instance in figure 3.17). Using a different time window, again with a variable number of oscillations, results change and lie in a gray-zone, where no decision can be taken ($\Delta\tau = 0.5 \div 0.6$; as for instance in figure 3.18). Therefore it is important to realize why the presence of stiction can be masked and not detected by the technique. This objective has been addressed by analyzing what happens for different extent of stiction and different process dynamics, including a

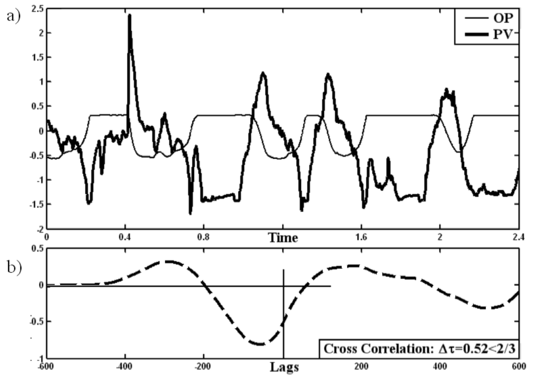


Figure 3.18: *Application of Cross correlation on data set* B*: a) PV and OP trends, b) Cross Correlation trend*

comparison with the other two techniques, Relay and Bicoherence.

3.4.1 Simulated data

Nominal case

The three techniques described in the previous section are compared by two different tests (process type $FOPTD$ and controller structure PI do not change):

- the control loop is subject to **stiction**: the stiction indexes ($\Delta\tau$, $\Delta\rho$ for the Cross Correlation technique; NGI and NLI for the Bicoherence technique; S_I for the Relay tecnique) are evaluated for different process conditions (varying the ratio θ/τ and the ratio $S/2J$);

- the control loop is subject to an **external perturbation**: the stiction indexes are evaluated varying the ratio θ/τ and the ratio ω_D/ω_U, where ω_D is the frequency of the recorded oscillation and ω_U is the ultimate frequency of the system.

Particularly, Cross Correlation technique was used to validate Karnopp and Choudhury models with a direct comparison of performances, as shown in figure 3.19.

Results are expressed in terms of the index $\Delta\tau$; (analogous results can be obtained using $\Delta\rho$). Stiction parameters are: the ratio between dynamic and static friction (F_D/F_S, for the Karnopp model, figure 3.19a) and the ratio between the dead zone and the jump ($S/2J$, for the Choudhury model, figure 3.19b). It can be seen that the presence of stiction is clearly indicated only for processes with large values of the ratio θ / τ and for stiction situations represented by large values of

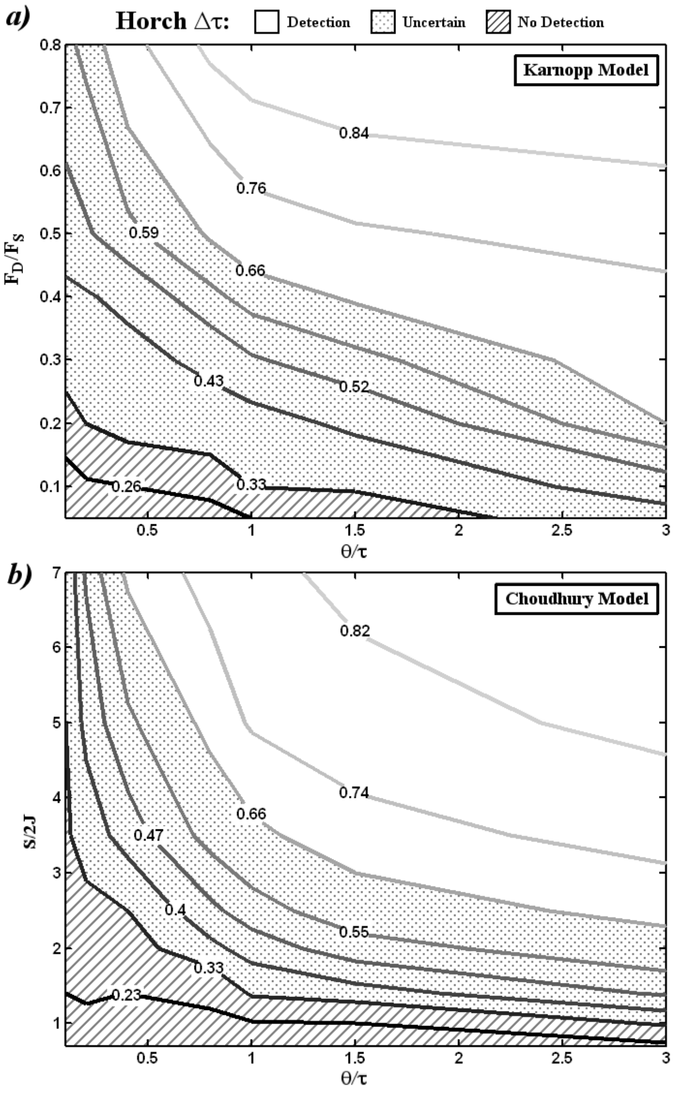


Figure 3.19: *Application of Horch technique; a): Karnopp model, b): Choudhury model*

F_D/F_S (or $S/2J$). On the contrary, for smaller values, no decision can be taken or stiction is not indicated, even though it is present.

A major point resides in similarities between Karnopp and Choudhury models, both indicating correctly white-zones (correct detection of stiction), gray-zone (uncertainty) and black-zones (wrong result). Distribution and extension of those three zones seem to confirm the possibility of simulating stiction with both models indiscriminately; in the sequel Choudhury model will be adopted.

Furthermore, it can be pointed out the effect of process dynamics: for low values of θ/τ (lag dominant processes), the presence of stiction can be hidden (or no decision can be taken), whatever the value of the stiction parameters. Applying the other two techniques a similar trend is obtained, in the sense that uncertainty regions appear for $\theta/\tau \ll 1$, even though they have a smaller extent (figure 3.20 for Bicoherence and figure 3.21 for Relay). All the techniques may fail for integrating processes; this is not surprising, as it can be seen as a limit case for $\tau \gg 1$. The previous mentioned technique [26] can be used, with possible drawbacks put into evidence in section 3.3. This result can be considered the first contribution of this study: indications deriving from different techniques are strongly affected by process dynamics, which can hide the presence of stiction.

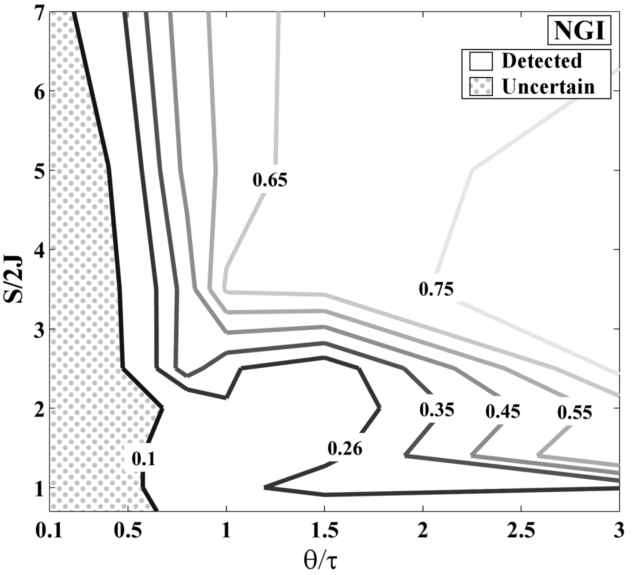


Figure 3.20: Simulation results for Bicoherence technique in case of stiction, by varying process and stiction parameters

In case of sinusoidal disturbances (figure 3.22), a range $\omega_D/\omega_U = 0.1 \div 10$ was investigated.

In this case, for all the three techniques, values of indexes do not change with the ratio θ/τ. The frequency of oscillations affects only Horch's technique [25], which can give uncertain results for low values of (ω_D/ω_U). For $\omega_D/\omega_U = 0.5, \Delta\tau \geq 1/3$ (uncertainty), while for $\omega_D/\omega_U = 0.2, \Delta\tau \geq 2/3$ (error in indicating stiction instead of sinusoids). This result can be considered implicit from [25], because a phase shift equal to $\pi/2$ between OP and PV (sinusoidal input and PI controller) is clearly observed only at high frequency ($\omega_D \ll \omega_U$), when the integral component dominates.

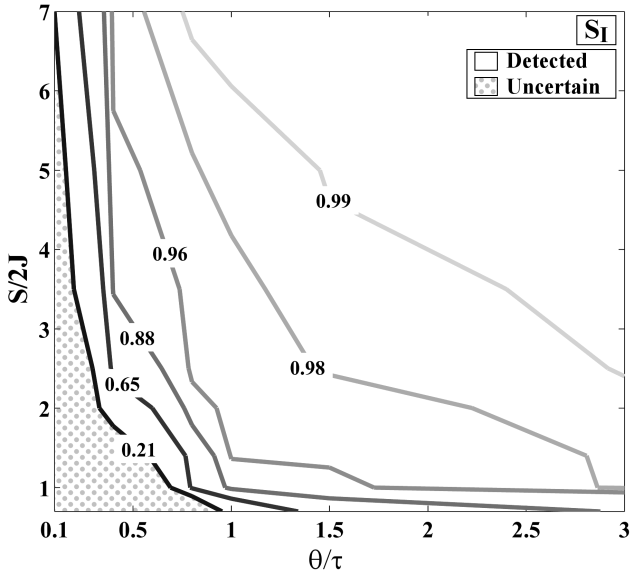


Figure 3.21: *Simulation results for Relay technique in case of stiction, by varying process and stiction parameters*

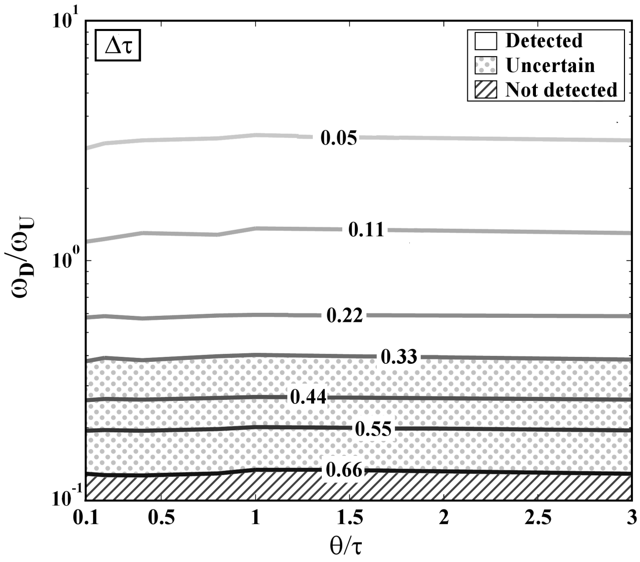


Figure 3.22: *Simulation results for Horch technique in case of sinusoidal disturbances by varying process and disturbance parameters*

Even though the practical consequence of this uncertainty in recognizing a low frequency sinusoid can be less severe, it must be pointed out that Relay and Bicoherence techniques are almost unaffected by it; in particular $S_I \in [-1; -0.9]$ and $NGI < 0$, $NLI \in [0.01; 0.08]$ in all the investigated frequency range, thus clearly indicating a sinusoidal signal.

The existence of uncertainty zones in stiction detection brings as a consequence for the Relay and the Bicoherence techniques that, without any further knowledge on the process, if stiction or oscillating disturbance are indicated, they are really present; an uncertainty region may hide the presence of stiction for lag dominant processes. To be recalled that the Cross Correlation technique may give wrong results in both cases, according to process parameters.

Robustness to noise

The same analysis was repeated adding a noise on the signal of PV and OP to investigate the robustness of the three techniques. A noise able to change the shape of the oscillation was created by varying its amplitude and period in three steps:

- adding a sinusoidal signal with a variation in the amplitude
- modifying the previous signal by the use of a first order filter with a variation in the time constant.
- overlapping a gaussian random noise

The algorithm receives as inputs PV values and desired ratio R between noise and signal amplitude; all variations depend on a randomly generated parameter $rand(x)$; the output is represented by PV_N, signal corrupted by noise. First step of the used algorithm is the evaluation of frequency of the oscillation ω_D, used to built a sinusoidal function with same amplitude and frequency of PV:

$$S_N^{(1)} = sin(\omega_D \cdot t_K) \cdot \max_t \{PV\} \cdot \delta(t) \qquad \text{(Eq. 4-8)}$$

where the suffix $S_N^{(x)}$ indicates the sinusoidal noise signal at step x of described algorithm. $\delta(t)$ is a function, whose value is updated randomly every half-cycle (detected by a change in PV sign):

$$\delta(t_K) = \begin{cases} rand(x) \cdot R & if \quad sign(PV(t_K)) \neq sign(PV(t_{K-1})); \\ \delta(t_{K-1}) & if \quad sign(PV(t_K)) = sign(PV(t_{K-1})); \end{cases} \qquad \text{(Eq. 4-9)}$$

With this definition of $\delta(t)$, the original signal is corrupted by a sinusoid composed by half-cycles with different amplitudes, randomly varied. In second step of the algorithm, time constant τ for the filtering action is calculated to obtain a phase shift $\Delta\phi = (\pi/\omega_D)R$. A parameter $a(t)$ is defined (Eq. 4-10) and a further corruption of signal is performed (Eq. 4-11); finally, in step three of the algorithm, a random noise is added by using a normal distribution $\psi(0, R)$ with zero mean and variance equal to R (Eq. 4-12):

$$a(t) = exp\left(\frac{-1}{\tau \cdot \delta(t)}\right) \qquad \text{(Eq. 4-10)}$$

$$S_N^{(2)}(t_K) = -a(t_K) \cdot S_N^{(2)}(t_{K-1}) + (1 - a(t_K)) \cdot S_N^{(1)}(t_{k-1}) \qquad \text{(Eq. 4-11)}$$

$$S_N^{(3)} = S_N^{(2)} + \psi(0, R) \qquad \text{(Eq. 4-12)}$$

The original signal is therefore corrupted by adding the sinusoid $S_N^{(3)}$ to the original signal: $PV_N = PV + S_N^{(3)}$. An example of the corruption of signal for different level of noise is depicted in figure 3.23 for different values of parameter R:

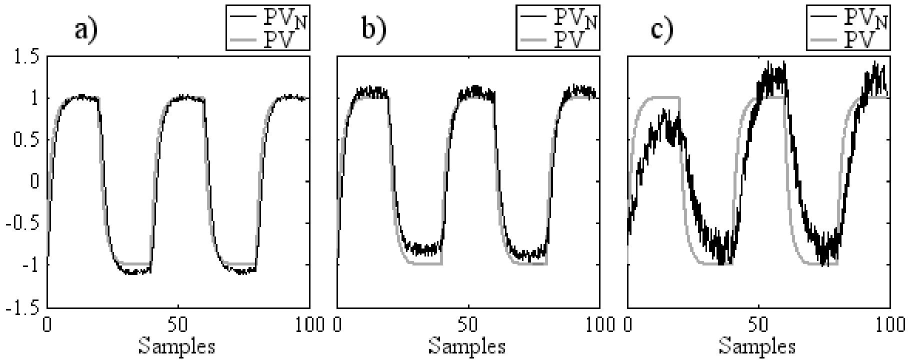


Figure 3.23: *Effect of simulated noise: a)* $R = 5\%$ *b)* $R = 15\%$*, c)* $R = 30\%$*; original signal PV (gray), corrupted signal PV*$_N$ *(black)*

It is evident that this noise is more severe than a gaussian noise and that, increasing the value of N, the corrupted signal becomes very different from the original, thus affecting the efficacy of the three techniques. Results are presented in figure 3.24.

To be noted that the effect of the noise is more evident for lower values of $S/2J$ and θ/τ; for this reason the zone of the correct detection of stiction decreases rapidly for the Cross Correlation, while for the other two techniques it can be considered that robustness is maintained up to a value of $N/A = 25\% \div 30\%$, which represents a large error on plant data.

3.4.2 Plant Data

To confirm previous indications obtained by simulation about capability of different techniques to detect stiction, data recorded from eight industrial loops of a refinery plant were analyzed.

All loops (no integrators were present) were known to be affected by stiction; recorded oscillation on $SP - PV$ and OP are illustrated in figure 3.25.

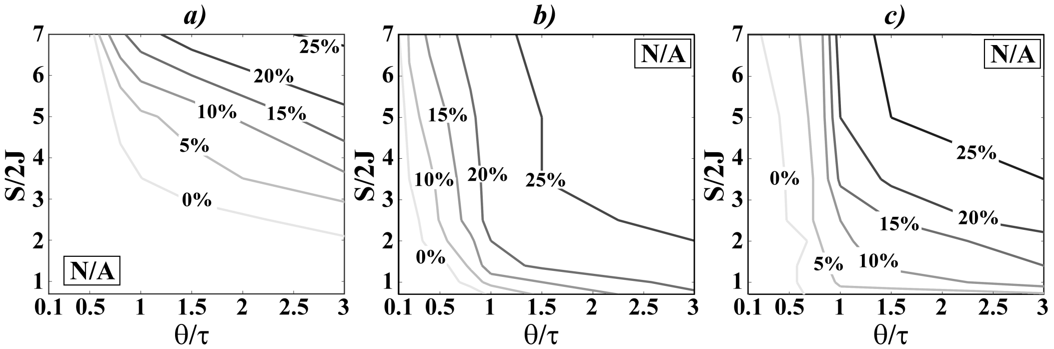


Figure 3.24: Shift of the zone of uncertainty increasing the ratio noise/signal (N/A) for: a) Cross Correlation, b) Bicoherence, c) Relay technique

It can be seen that loops from 1 to 5 show very regular and clean signals (almost no noise); loops 6 is characterized by noise and irregular amplitude, caused by valve saturation; in loop 8, the amplitude is small and the ratio noise to signal is high. Loop 7 has an intermediate behavior (low noise, but less regular oscillations). For each loop, more than 6000 samples are collected, but in figure 3.25 abscissas are chosen to show clearly the shape of oscillations.

This analysis of data was also oriented to investigate practical issues, both for on-line and off-line implementation. Among them: the effect of number of oscillations and of chosen time window on the efficiency of the techniques, related computation times and ease of application. For this reason, a total of 30 half-cycles was then analyzed for each loop, by grouping data in 4 different sets made respectively by 6 time windows of 5 half-cycles, 3 of 10, 2 of 15, 1 of 30. The minimum number of oscillations was found to be equal to 5 for the Bicoherence and the Relay techniques: by performing the analysis on a smaller number of cycles, results may depend on the chosen time window. On the contrary, for the Cross Correlation technique in most cases two cycles can be enough.

About duration of elaboration (on a 1 GHz - Pentium IV processor): the Cross-Correlation resulted to be the fastest (negligible time also for the largest set of data); the Bicoherence times requires from few seconds to less than one minute, while the Relay shows longer times (up to 4 minutes for the largest set). The ease of implementation is also favorable for the Cross-Correlation technique (few lines of any programming language), while the other two require more complex procedures (in this case Matlab built-in functions were used).

About efficiency of detection, results are reported in table 3.2 for the three different techniques. “Yes” indicates a correct detection of stiction, “No” an incorrect interpretation and “Unc” stands for uncertainty. Where only one answer appears, it means that it was confirmed from all the 4 sets of data; where two answers appear, it means that results changed with the observed set of data.

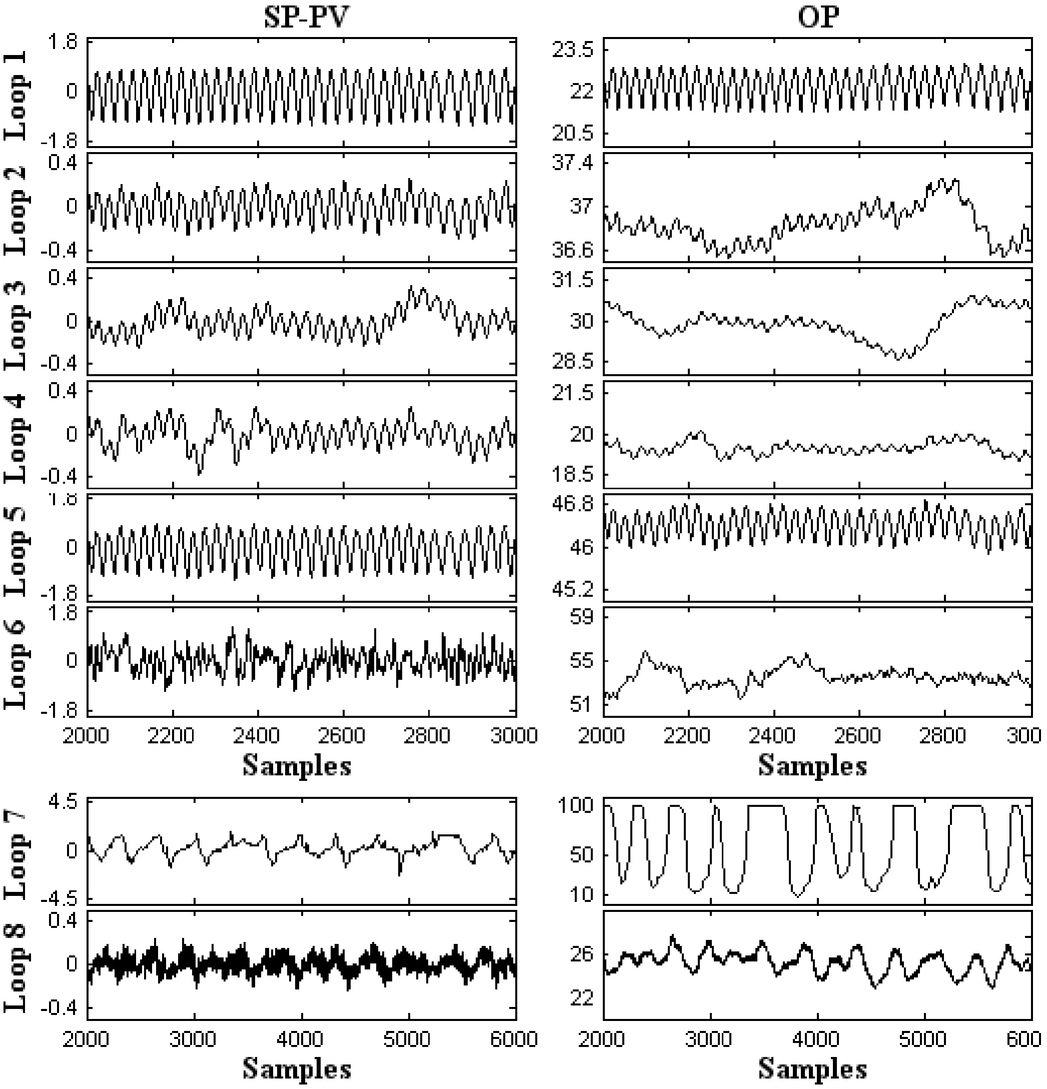


Figure 3.25: *Recorder values of $SP - PV$ and OP, as function of time samples, for eight industrial loops in presence of stiction*

Table 3.2: *Results from the application of the three techniques*

Loop	1	2	3	4	5	6	7	8
C_{XY}	No	Yes	No	Yes	No/Unc	Yes/Unc	Yes	Yes
bic^2	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes/No
$Relay$	Yes	Yes/Unc	Yes	Unc	Yes	Yes	Yes	Unc

As synthesis, about the efficiency of different techniques in this application to industrial data:

- Cross Correlation detects successfully the stiction on 4 loops, there is a conflict for for 2 loops and is not able to detect stiction on 2 loops, imputing the cause to an external disturbance.

- Bicoherence detects correctly the stiction on 7 loops with a conflict on the last loop.

- Relay detects stiction on 5 loops; on the remaining 3 loops gives 2 uncertainties and 1 conflict.

Therefore the Bicoherence technique shows its superiority in detecting the presence of stiction. From a closer examination of output results, the Relay technique allows to detect the presence of asymmetric stiction in loop 1 and 5. The phenomenon depends on the fact that in these two cases stiction is present only for positive values of PV (valve opening, or viceversa, according to gain sign).

Two possibilities can be suggested to overcome the cases of uncertainty in taking decisions:

1. By a simple plant test, i.e. decreasing the controller gain K_C, (MV not available). Referring to figure 3.10, the frequency of the oscillating disturbance does not vary (the controller parameters influence only amplitude and phase), while in case of stiction, a significant decrease in the frequency of the recorded oscillation is observed.

2. By requiring the availability of MV, as variable to be recorded in the data acquisition system; in this case, the typical characteristic of stiction is maintained, thus allowing its detection [34]. For instance, a direct test on the trend of $MV = f(OP)$, would be able to detect the presence of stiction in a faster and more efficient way; this is shown in figure 3.26 for loops 5 and 7 (flow control loops: $MV = PV$): the characteristic limit cycle is evident, indicating without doubt the presence of stiction.

3.5 Stiction index

Once stiction is detected, the procedure must indicate to the operator the strategy to be implemented choosing between maintenance of the valve and use of compensators (as described for instance in [17] or [18]). To built a performance index to distinguish between the two cases, characteristic oscillations of OP variable are analyzed. Similarly to Relay technique, each half-cycle is examined, defining two values: IAE_{OP}, integral of absolute value of error of the half-cycle, and IAE_{MAX},

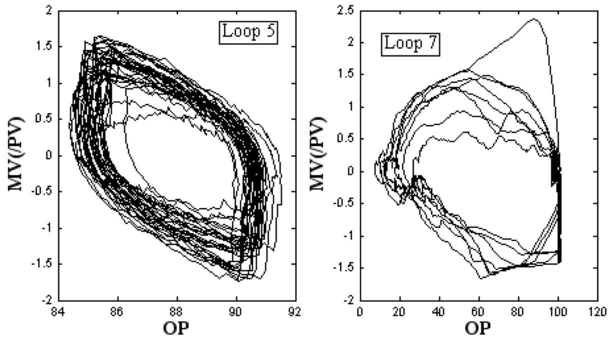


Figure 3.26: *Examples of $MV = PV = f(OP)$ plots for the two flow controllers in loop 5 and 7; presence of stiction*

the maximum value that IAE_{OP} would assume in case of permanent blockage of the valve. IAE_{MAX} can be evaluated as a function of $OP_{\%,M}$, mean value of controller output on a $[0 \div 100]\%$ scale and T_{OSC}, duration time of an half-cycle (Eq. 4-13). On the basis of IAE_{MAX} and IAE_{OP} a new index ST_{EST} can be defined (Eq. 4-14), thus estimating quantitatively stiction and helping operators in choosing the most appropriate intervention.

$$IAE_{MAX} = T_{OSC} \cdot min \left\{ OP_{\%,M}; 100\% - OP_{\%,M} \right\} \qquad \text{(Eq. 4-13)}$$

$$ST_{EST}^{(i)} = \frac{IAE_{OP}^{(i)}}{IAE_{MAX}^{(i)}} \qquad \text{(Eq. 4-14)}$$

$ST_{EST}^{(i)}$ index, evaluated for i^{th} half-cycle, lies in range $[0 \div 1]$; the larger its value, the bigger the influence of stiction on collected variables. Using a conservative approach, the procedure considers as reference the maximum value among indexes obtained from analysis of all collected half-cycles (Eq. 4-15):

$$ST_{EST}^{MAX} = \max_{i=1..N} \left\{ ST_{EST}^{(i)} \right\} \qquad \text{(Eq. 4-15)}$$

for N collected half-cycles. The definition of a threshold value is based on following considerations:

- OP variable shows for all cases of stiction an oscillation, which has an approximatively triangular wave shape

- it is assumed that stiction can be considered tolerable (use of compensators) when, because of its presence, oscillations on OP do not reach saturation conditions

A value of $ST_{EST}^{MAX} = 0.5$ derives from these two conditions, for lower values compensation is the suggested solution, for larger values maintenance. Trend of ST_{EST}^{MAX} as function of stiction parameters (S, J) and process parameters (θ, τ) is shown in figure 3.27.

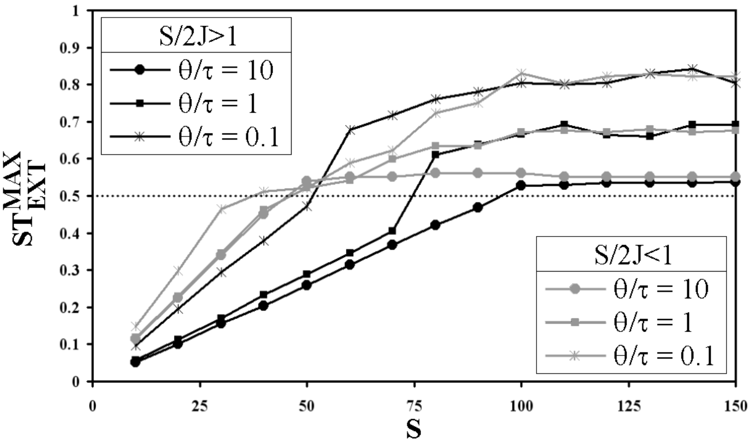


Figure 3.27: *Trends of ST_{EXT}^{MAX} as function of stiction (S, J) and process (θ, τ) parameters*

A valve in saturation conditions is associated to all points, whose index is larger than 0.5; to be noted that this phenomenon is shown for different values of S at different values of the ratio θ/τ. As a matter of fact, reaching saturation conditions depends on stiction parameters, but also on active force generated by controller, which is function of its parameters gain and integral constant. Furthermore, when $S/2J < 1$, saturation conditions are reached more easily because of a strong effect of stiction (large jump of the valve when the motion starts). For all cases, when the valve is not saturated, the proposed index varies linearly with S, to reach an asymptote, whose value depend by process dynamic (θ/τ parameters).

This analysis demonstrates the efficacy of the index, which is able to indicate correctly when maintenance is requested. ST_{EXT}^{MAX} can be used also for analysis of historical data about stiction phenomena: repeating the index evaluation in different dates and collecting its values, it is possible to study the evolution of stiction. Furthermore, an interpolation of data gives the possibility to estimate the time required by the valve will to reach the saturation and, as consequence, to plan maintenance of valves.

Chapter 4

Identification and Retuning

Previous chapters report a methodology to detect poor performing loops: by applying a classification procedure, the presence of large errors due to improper controller tuning can be indicated, considering both sluggishness and aggressiveness. Whatever the case, a procedure to calculate new tuning parameters will be described in this chapter with the aim of finding a balance between speed in rejecting disturbances and robustness to uncertainties.

Tuning of industrial controllers is very often far from optimal, because of the general philosophy to avoid oscillations (synonym of instability) and therefore to apply conservative tuning which causes slow responses [37]. This is also a consequence of the fact that standards about the tuning procedure are not strictly enforced on many plants and left to operators skill. Moreover, even a perfect tuning may become inadequate in the case of changes in operating condition or in the presence of external perturbations.

For plant subject to frequent changes in operating conditions, there are advantages in introducing changes in controller parameters, as they can be kept closer to optimal values. Autotuning techniques have this aim, but in their most common application they require to introduce explicit perturbations in the plant [38], [33], [5]. This is certainly a drawback, even though the amplitude of the perturbation (step, relay tests, etc.) can be limited to a minimum, depending on the plant noise. Also, according to plant dynamics, the duration of the experiments is not negligible in many cases and this is an important parameter to take into consideration, especially for techniques which require repetition of experiments to improve process identification [5].

Nowadays, data acquisition systems give the availability of very large amount of plant data and this allows the user to assess controller performance from the response to an external perturbation entering the plant, without the need of explicit

[*] Portions of this chapter are published in [7],[35],[36]

solicitations. In principle it is possible to identify both process and disturbance dynamics, by analyzing values of controlled variable, controller output and set-point value, collected during routine operation. Required additional information are the controller parameters and the sampling time of collected data, usually available.

The developed procedure is based on three steps: identification of processes and disturbances with models, calculation of tuning parameters with different techniques, evaluation of the performance the loop would reach adopting new proposed controllers.

4.1 Identification of Process and Disturbance

The reference scheme for a Single Input Single Output system is here recalled (in figure 4.1), representing P the process, P_D the disturbance and C the controller. Having excluded static friction problems on the actuator, the valve dynamic will be neglected in this analysis.

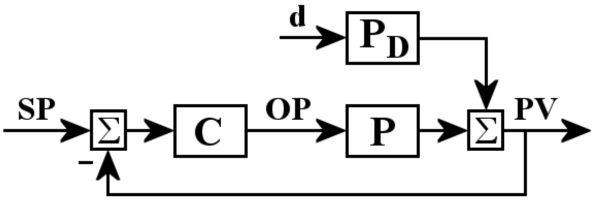


Figure 4.1: Reference scheme for a Single Input Single Output ($SISO$) system

To evaluate the effect of disturbance d and control action OP on the controlled output PV, two basic equations can be written (Eq. 5-1 and Eq. 5-2).

$$PV = \frac{P_d}{1 + P \cdot C} \cdot d \qquad \text{(Eq. 5-1)}$$

$$PV = P_d \cdot d + P \cdot OP \qquad \text{(Eq. 5-2)}$$

4.1.1 Linear Least Square technique

In the deterministic case, d can be assumed as a generic step disturbance (i.e. constant); $u = u(t)$ is available as a measurement. If a process and disturbance have the same dynamics, equation Eq. 5-2 can be simplified. A linear least square (LLS) technique can be used to solve the problem of finding a model which minimizes the mean square error (MSE) between collected (PV) and approximated data ($\widetilde{PV}$):

$$MSE = \frac{\sum_{i=1}^{N}(PV_i - \widetilde{PV}_i)^2}{N} \qquad \text{(Eq. 5-3)}$$

The relationship between inputs (OP,d) and output PV in discrete form for a second order plus time delay ($SOPTD$) process in case of equal dynamic but with different gains is:

$$PV(z^{-1}) = \tilde{P} \cdot OP(z^{-1}) + \tilde{P}_d \cdot d(z^{-1}) \qquad \text{(Eq. 5-4)}$$

where:

$$\tilde{P} = \frac{b_1 \cdot z^{-1} + b_2 \cdot z^{-1}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2}} \cdot z^{-q} \qquad \text{(Eq. 5-5)}$$

$$\tilde{P}_d = \frac{c_1 \cdot z^{-1} + c_2 \cdot z^{-2}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2}} \qquad \text{(Eq. 5-6)}$$

.

Eq. 5-4 can be translated for the generic sample k:

$$\begin{aligned} PV_k = &- a_1 PV_{k-1} - a_2 PV_{k-2} + b_1 OP_{k-1-q} + b_2 OP_{k-2-q} + \\ &+ c_1 d_{k-1} + c_2 d_{k-2} \end{aligned} \qquad \text{(Eq. 5-7)}$$

with k ranging from $k = 3 + q$ (to start with the first sample OP_1), to $k = N$; $q = \theta / t_S$ is the ratio between time delay and sampling time. The problem can be stated as:

$$\hat{PV} = A \cdot \Gamma \qquad \text{(Eq. 5-8)}$$

where $\hat{PV}$, is the vector of process output data from sample $q + 3$ to sample N (Eq. 5-9); A is the array of collected data (both PV and OP), having for the chosen model form the following structure with 6 columns and $N - 2 - q$ rows (Eq. 5-10), Γ is the vector of the 6 model parameters (Eq. 5-11).

$$\hat{PV} = \left[PV_{q+3}, PV_{q+4}, \ldots, PV_N\right]^T \qquad \text{(Eq. 5-9)}$$

$$A = \begin{bmatrix} PV_{q+2} & PV_{q+1} & OP_2 & OP_1 & d_2 & d_1 \\ PV_{q+3} & PV_{q+2} & OP_3 & OP_2 & d_3 & d_2 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ PV_{N-1} & PV_{N-2} & OP_{N-1-q} & OP_{N-2-q} & d_{N-1-q} & d_{N-2-q} \end{bmatrix} \qquad \text{(Eq. 5-10)}$$

$$\Gamma = \left[-a_1, -a_2, b_1, b_2, c_1, c_2\right]^T \qquad \text{(Eq. 5-11)}$$

In this form, the problem is non linear because of the presence of the term q, unknown, being a function of process time delay. To solve it with a LLS technique, values of q are assumed and consequently the Γ array is calculated by:

$$\Gamma = (A^T A)^{-1} \cdot A^T \cdot \hat{PV} \qquad \text{(Eq. 5-12)}$$

Computations are repeated for different values of q, starting from 1 and finishing with $N-6$, upper limit which guarantees a unique solution to the LLS problem (greater values will result in a A matrix with less than 6 rows, thus not allowing the system to evaluate Γ). This brings to a family of $N-6$ models used to generate approximate output values; the model which gives the lower value of MSE is chosen among them. Even tough the large number of LLS application for different q, calculation time is almost negligible, being based only on matrix inversions: generally of a magnitude order of 20 seconds.

Henceforward, the assumption $P = P_d$ allows an analytical solution to the problem and this is certainly an advantage with respect to optimization algorithms, which can bring to local minima; nevertheless this assumption can have a drawback, with a strong effect on the quality of results, as explained in the sequel.

4.1.2 Simplex method algorithm

A new method is proposed for the case of different dynamics for P and P_d. The recorded set of data PV is fitted by means of a second order plus time delay ($SOPTD$) models, here represented as continuous transfer functions, in order to put into evidence the physical meaning of parameters :

$$\widetilde{P} = \frac{K \cdot e^{-\theta s}}{\tau^2 s^2 + 2\tau\xi s + 1}; \qquad \widetilde{P}_d = \frac{K_d \cdot e^{-\theta_d s}}{\tau_d^2 s^2 + 2\tau_d \xi_d s + 1}; \qquad \text{(Eq. 5-13)}$$

With this choice of models it is possible to approximate both lag and time delay dominant cases, as well as overdamped or underdamped and higher order responses. The identification problem can be stated in finding the vector of models parameters $\phi = [K, \tau, \xi, \theta, K_d, \tau_d, \xi_d, \theta_d]$ which generates the best fit of collected data, by minimizing the mean square error (MSE).

The identification procedure is based on a modified Simplex Method, originally proposed by Nelder and Mead [39] and described in all analytical details in [40], where default values for some of the parameters of the original method are suggested (more details in Appendix B). Adopting the simplex method, convergence is guaranteed, but some strategic choices are required to reduce the possibility for the algorithm to be stuck in local minima. For this reason the identification procedure is divided into three steps:

1. research of a suitable initial point to start the algorithm;

2. application of the simplex method for a fixed value of K_d, as it was found that this has a positive effect in reducing the number of local minima;

3. refinement of results by repeating the simplex algorithm on all the variables contained in the vector ϕ, including also K_d.

As first step, the starting point ϕ_0 is calculated from collected data and a grid around this point is built; initial values and nodes of the grid are chosen as follows:

- **gains:** K_d is calculated to obtain $max(PV) = max(\widetilde{PV})$; K is roughly estimated by considering that the steady state of the control action $u_\infty = -K_d/K$ and spanned around this value, being K_d unknown.

- **time delays:** θ_d can not be evaluated from collected data, for this reason is initially assumed equal to zero and changed in the next steps. On the contrary an overestimation of process delay can be taken as: $\theta = t_{MAX}$, time in which the process output Y reaches its maximum value.

- **dynamic elements:** as rough estimation, τ_d and τ assumed equal to t_{MAX} and to the settling time of the disturbance rejection, respectively; ξ and ξ_d are initially assumed equal to 1.

The initial estimates of parameters are used to build the initial grid; the point of the grid which generates the minimum MSE is chosen as starting point for the simplex algorithm.

As second step, the simplex method is used to solve the following sub-problem:

$$\min_{\phi}\{MSE\} \quad s.t. \begin{cases} K_d = \frac{max(Y)}{max(\widetilde{Y})} \\ min\{\tau, \tau_d, \theta, \theta_d, \xi, \xi_d\} > 0 \end{cases} \qquad \text{(Eq. 5-14)}$$

The first constraint assures that approximated and recorded data have the same maximum; the second constraint guarantees that the parameters are positive, as required by the physical consistence of process and disturbance dynamics. The problem is solved by means of the *fminsearch* MATLAB function, with some modifications to take into account the presence of constraints, not considered in the original Simplex algorithm (Appendix B for more details).

In the third step of the procedure, the first of the two constraints in equation Eq. 5-14 is released (i.e. also K_d can vary) and the modified simplex method is applied again using as starting point the minimum found at the previous step. This procedure is certainly more complex than the LLS algorithm and generally requires longer computation times (about 200 seconds); however times are negligible if compared with typical duration of data acquisition for industrial plants. Anyway, the key point is that the possibility of taking into account different P and P_d dynamics can give better results in identification and closed loop performance. This issue will be discussed in the sequel.

4.2 Controller design

Once the identification is performed, the design of the new controller becomes the next step of the procedure. The design of the controller can be different according to the acquired knowledge on the process: from the simplest Ziegler-Nichols tuning rule [13] (based on minimum information), to smart tuning rules (for instance those proposed by Skogestad [41], which require an approximated model), to model based techniques (which have the objective of minimizing a function of the error $e = r - Y$). In this study, controllers having PI structure, tuned according to [41] (equation Eq. 5-15; K_C: controller gain, τ_I: integral time) or designed for a minimum $ITAE$ performance objective (equation Eq. 5-16), will be compared, as examples of different approach to control design:

$$\begin{array}{l} K_c = \frac{0.5\cdot\tau}{K\cdot\theta} \\ \tau_I = min\{\tau, 8\theta\} \end{array} \qquad \text{(Eq. 5-15)}$$

$$\underset{K_c,\tau_I}{min}\; ITAE = \underset{K_c,\tau_I}{min} \int |e(t)| \cdot t \cdot dt \qquad \text{(Eq. 5-16)}$$

Whatever the design technique, the goal of identification for control purposes is to obtain good closed loop performance; its effectiveness has been evaluated by means of the $DITAE$ index, defined as the Difference between $ITAE$ of different controllers:

$$DITAE = \frac{|ITAE_M - ITAE_P|}{ITAE_P} \qquad \text{(Eq. 5-17)}$$

Suffix $\{\cdot\}_M$ indicates the controller designed on the model, but applied in the real plant; suffix $\{\cdot\}_P$ the controller (designed by the same technique) based on a perfect knowledge of the process. $DITAE$ is a synthetic indication of goodness of identification for control purposes: values close to zero indicates a correct result, while larger values corresponds to larger error in modelling. It should be noted that this index is also an implicit measure of robustness: in case of unstable controllers for the real process $DITAE \rightarrow \infty$, because of the very high value assumed by $ITAE_M$.

To examine $DITAE$ index, a basic case of First Order Plus Time Delay ($FOPTD$) process and disturbance in closed loop with a PI controller will be considered:

$$P = P_d = \frac{1 \cdot e^{-10s}}{10s + 1} \qquad \text{(Eq. 5-18)}$$

$DITAE$ is reported in figure 4.2 as a function of controller parameters (gain K_C and integral time τ_i).

Similar trends can be obtained varying process and disturbance structure and parameters; $ITAE_P$ corresponds to the minimum value of the function, where

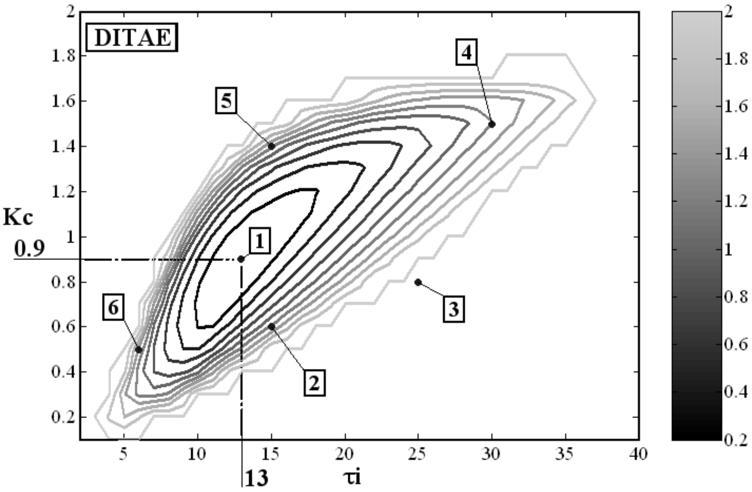


Figure 4.2: $DITAE$ *index, function of controller gain* K_C *and integral time* τ_i

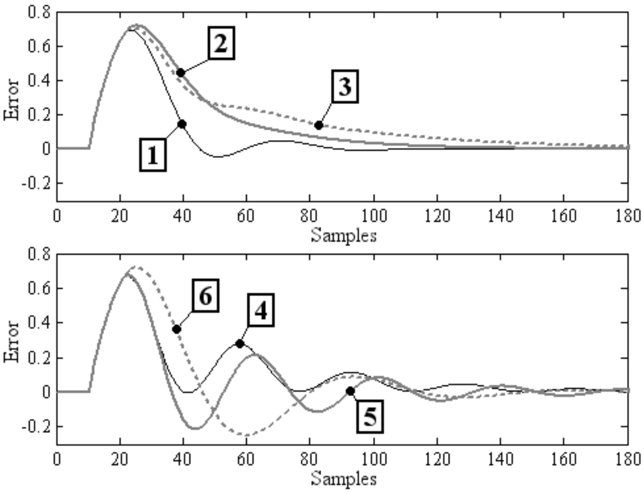


Figure 4.3: Examples of different responses, corresponding to controller parameters shown in figure 4.2

$DITAE = 0$. Whenever different controller parameters are adopted $DITAE$ increases because of sluggish or oscillating responses (figure 4.3).

4.3 Performance Evaluation

Different indexes are used to evaluate performance of designed controllers; because of lack of knowledge on real plant dynamic, the models (obtained as described in previous sections) must be used. Two indexes were taken from literature: to evaluate the number of significant oscillations of the disturbance rejection response see [1]; to calculate its sluggishness see [42]. The first technique is described in chapter 2 (sections 2.1.1, 2.1.2, and 2.2), there used to detect poor performing loops. On the contrary, the index for detection of sluggishness, named Idle Index I_I [42], will be described more in detail in this chapter. A third index is proposed, named $ITAEr$ [7] (meaning $ITAE$ Ratio): its purpose is to give a criterion to compare two controllers and particularly to evaluate the improvement in performance due to a new tuning with respect to the implemented controller.

4.3.1 Idle Index

Only collected data about PV and OP are sufficient for the application of this index in the nominal case (i.e. without either plant or simulated noise). However, being I_I projected for the case of disturbance rejection, also SP values must be collected to avoid the analysis of set-point tracking responses. Basically, first derivatives of OP and PV must be considered and the sign of their product evaluated: as example of application the study case described by Eq. 5-19 can be considered and two different controllers adopted:

$$\begin{aligned} C_1 &= 0.92 \cdot \frac{25s+1}{25s} \\ C_2 &= 0.98 \cdot \frac{12s+1}{12s} \end{aligned} \qquad \text{(Eq. 5-19)}$$

Responses with the chosen controllers and the sign of the product between $\partial PV/\partial t$ and $\partial OP/\partial t$ are depicted in 4.4.

In details, both the sluggish (figure 4.4a) and the fast response (figure 4.4b) have a starting phase where $\partial PV/\partial \cdot \partial OP/\partial t < 0$: the controller action does not affect the response before the time delay of the process. Differences appear after this period, when the two curves decrease with longer or shorter periods where $\partial PV/\partial t \cdot \partial OP/\partial$ is positive or negative. Particularly, for sluggish control action a prevalence of $\partial PV/\partial t \cdot \partial OP/\partial > 0$ is present; instead, for faster controllers alternate periods of time with different sign are shown: this example indicates a correspondence between the $\partial PV/\partial t \cdot \partial OP/\partial$ sign and sluggishness of controller.

The effect of the sign of the product of derivatives depends on the process gain sign; in this section the case of positive gain will be considered (for the opposite case, inverse relations must be adopted). Two parameters are defined, namely $T^{(+)}$ and $T^{(-)}$ as described by following equations:

$$T^{(+)} = \begin{cases} T^{(+)} + h & if \;\; \frac{\partial PV}{\partial t} \cdot \frac{\partial OP}{\partial t} > 0 \\ T^{(+)} & if \;\; \frac{\partial PV}{\partial t} \cdot \frac{\partial OP}{\partial t} < 0 \end{cases} \qquad \text{(Eq. 5-20)}$$

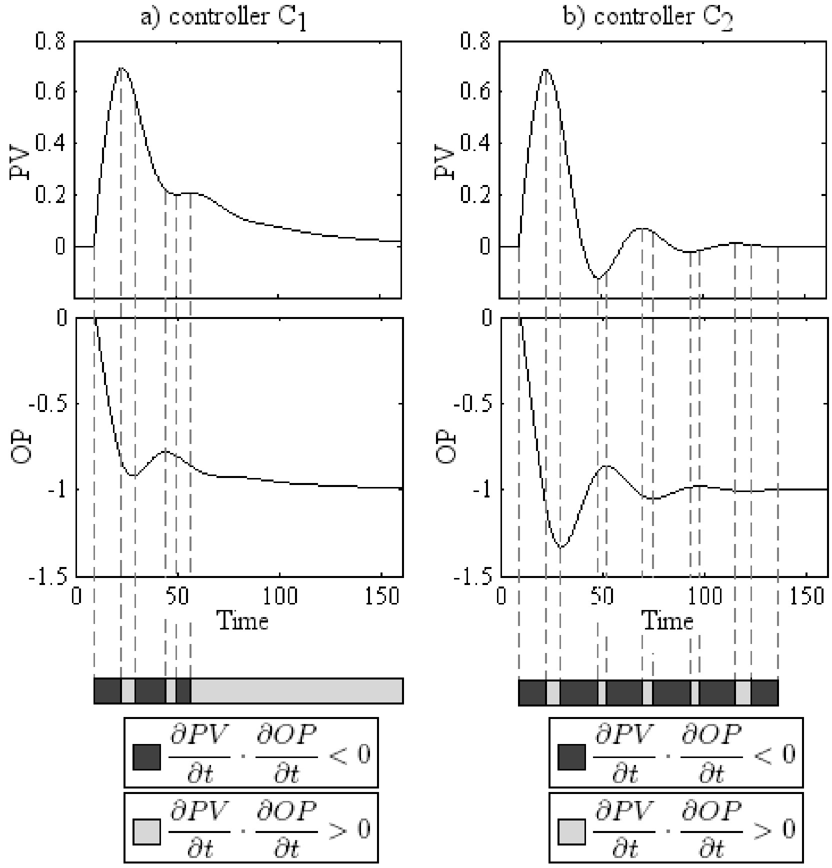


Figure 4.4: *Examples of different speed in rejecting disturbances: a) sluggish controller, b) faster controller*

$$T^{(-)} = \begin{cases} T^{(-)} + h & if \quad \frac{\partial PV}{\partial t} \cdot \frac{\partial OP}{\partial t} \leqslant 0 \\ T^{(-)} & if \quad \frac{\partial PV}{\partial t} \cdot \frac{\partial OP}{\partial t} \geqslant 0 \end{cases} \qquad \text{(Eq. 5-21)}$$

where h is the sampling time. These two parameters can be used to define the Idle Index:

$$I_I = \frac{T^{(+)} - T^{(-)}}{T^{(+)} + T^{(-)}} \qquad \text{(Eq. 5-22)}$$

The index lies in the range $[-1 \div +1]$, being $T^{(+)}$ and $T^{(-)} \in [0, +\infty]$. For sluggish responses, where $T^{(+)}$ prevails over $T^{(-)}$, $I_I \rightarrow +1$, for fast responses $I_I \rightarrow -1$.

Neglecting the knowledge on sign of process gain, which can be easily guessed by analyzing controller gain, a much subtle problem arises from data corrupted by noise. Derivatives can be strongly affected by this phenomenon; the implementation of filters would only change the response dynamic. Henceforward, I_I has to be evaluated on data obtained by simulations where models of process and disturbance are used, (i.e. the application of the index has to be done after the identification procedure).

4.3.2 ITAEr index

In case of a perfect control (minimum variance controller [43]) an error in rejecting disturbance is however present: during the period of time correspondent to process delay θ the controller action does not influence PV trends. After this period the controller completely counteract the disturbance, hence $SP(t)-PV(t) = 0\ \forall t > \theta$. As consequence a minimum error in rejecting disturbances can be evaluated as:

$$ITAE_{Lim} = \int_{t=t_0}^{t=t_0+\theta} |e(t)| \cdot t \cdot dt \qquad \text{(Eq. 5-23)}$$

The nearer to this limit the $ITAE$ value of the controller whose performance is evaluated, the better. $ITAEr$ definition is reported in Eq. 5-24.

$$ITAEr = \frac{ITAE_M\left(\tilde{P}\right) - ITAE_{Lim}}{ITAE_P\left(P\right) - ITAE_{Lim}} \qquad \text{(Eq. 5-24)}$$

where $ITAE_P(P)$ is the $ITAE$ value of collected data, namely a measure of performance of the controller used on real plant P, and $ITAE_M(\tilde{P})$ is the $ITAE$ value associated to the newly tuned controller, whose performance is calculated on the model of the process $\tilde{P}$, being impossible to evaluate its behavior on real plant until its implementation. It is evident that also for $ITAEr$ index a model of the process and disturbance is necessary; its application requires the implementation of an identification and retuning procedure.

$ITAEr$ index lies in the range $[0 \div +\infty]$: if, thanks to new tuning parameters, a minimum variance controller is obtained, therefore $ITAE_M \equiv ITAE_{Lim} \Rightarrow ITAEr = 0$. Excluding the case of a minimum variance controller, if the performance obtained with the new tuning improves, the index lies in the range $[0 \div 1]$; on the contrary a deterioration in performance is characterized by $ITAEr > 1$. In case of a minimum variance controller already implemented on plant $ITAEr$ can not be evaluated, being $ITAE_P \equiv ITAE_{Lim}$ and the denominator in Eq. 5-24 equal to zero. This case is not significant and easily detectable by Idle index, in this case assuming exactly a -1 value.

4.4 Simulation results

The identification procedure was applied on the following different types of processes and disturbances:

$$\begin{array}{l} FOPTD: P=\frac{K\cdot e^{-\theta s}}{\tau s+1}; \quad P_d=\frac{K_d\cdot e^{-\theta_d s}}{\tau_d s+1} \\ SOPTD: P=\frac{K\cdot e^{-\theta s}}{\tau^2 s^2+2\tau\xi s+1}; \quad P_d=\frac{K_d\cdot e^{-\theta_d s}}{\tau_d^2 s^2+2\tau_d\xi_d s+1} \\ HOPTD: P=\frac{K\cdot e^{-\theta s}}{(\tau s+1)^m}; \quad P_d=\frac{K_d\cdot e^{-\theta_d s}}{(\tau_d s+1)^{m_d}} \end{array} \qquad \text{(Eq. 5-25)}$$

By appropriate variations of the parameter values it is possible to both cover a large range of dynamics (overdamped and underdamped, lag and delay dominant) and analyze situations with a different structure between P and P_d.

Data sets are generated by recording time values of the $PV$and OP variables after a step disturbance enters a loop with an active initial PI controller, tuned according to the Ziegler-Nichols technique: these data can be viewed either as resulting from the adoption of a sub-optimal controller or from an optimal controller designed for a different process, the deterioration in performance being due to a change in the process itself. In both cases, the initial situation represents a sub-optimal controller: application of the identification algorithms renders a new model available and results in the design of a new controller.

In the case of equal dynamics, different values of K and K_d gain do not affect the accuracy of either identification technique. LLS can be formulated (see previous sections) such that an exact analytical solution (in terms of LLS) is obtained. For the particular case of $FOPTD$ process and disturbance, the relative error on parameters value is less than 0.01%, leading to absolutely negligible $DITAE$ errors.

The accuracy of the SPX algorithm is slightly more affected: in the same example, the error on values of identified parameters is always less than 4% (maximum in simulations 3.8%). Values of $DITAE$ are less than 10% (maximum in simulations 8.2%), to indicate good closed loop performance (PI controllers tuned according to Skogestad tuning rule [41] have been adopted). Almost overlapping responses are obtained with controllers based on the identified model or on the real process, as depicted in 4.5.

To verify the applicability of the simplex technique, the effect of different P and P_d dynamics has also been analyzed; again, a comparison with LLS has been performed. In this case, a controller designed according to a minimum $ITAE$ performance objective was chosen for both identification techniques. Intensive simulations have been carried out for a total of around 400 cases with: $\theta, \tau, \theta_d, \tau_d = 1 \div 100$, $\xi, \xi_d = 0.1 \div 2$ and $m, m_d = 3 \div 5$. As general rule, it can be said that for different process and disturbance dynamics, a general result shows that the simplex technique is able to provide good identification and closed loop responses, while LLS is largely affected by differences between P and P_d.

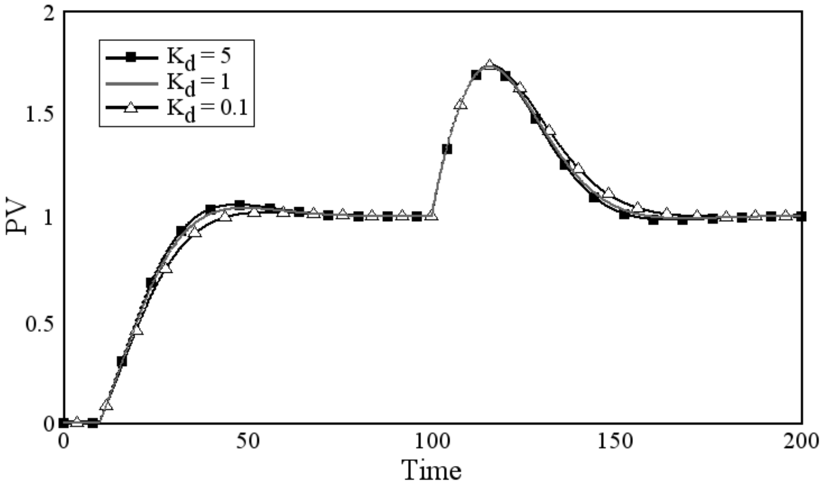


Figure 4.5: *Results of the application of Simplex method with different K_d gains*

The result of the particular case of a $FOPTD$ process and disturbance, with $\theta = 10$ is illustrated in figure 4.6a) and 4.6b).

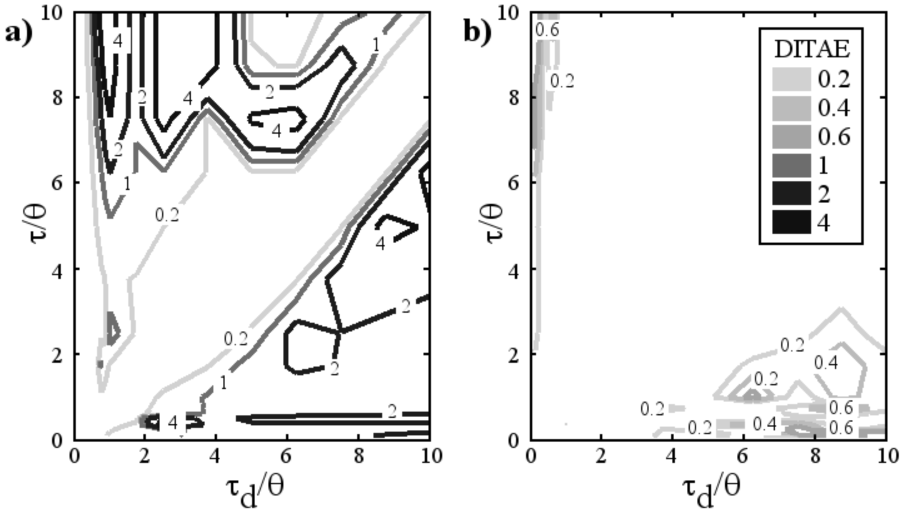


Figure 4.6: *Maps of $DITAE$ values, as a function of the main parameters for the case of $FOPTD$ process and disturbance ($\theta = 10$): a) LLS technique; b) Simplex Method*

The LLS technique only shows small $DITAE$ values for cases on the diagonal (i.e.: $\tau_d = \tau$ to indicate equal process and disturbance); but in all other cases, $DITAE$ values increase rapidly to indicate an evident deterioration of identification, which combines with poor performance in the real process; sometimes instability conditions may occur. It should be pointed out that, according to its definition,

$DITAE$ values larger than 1 are associated with poor performance deriving from identification error.

On the contrary the simplex method is able to build a correct model for P and P_d separating their effects on the recorded disturbance rejection curve, as indicated by much smaller off-diagonal $DITAE$ values.

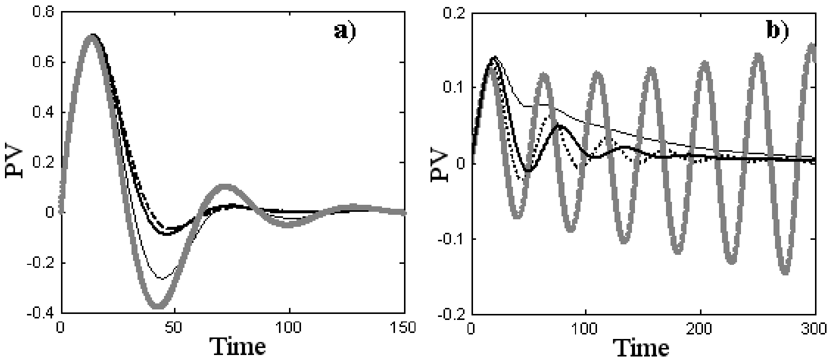


Figure 4.7: *Application of Simplex and* LLS *technique: a)* $\tau = 50$, $\tau_d = 10$, *b)* $\tau = 25$, $\tau_d = 100$; *initial controller (thin black), optimal controller by* LLS *(thick grey), by Simplex (thick black), optimal controller based on real process (dots).*

An example of a comparison between SPX and LLS is shown in figure 4.7. In the first case (a), the scarce identification provided by the LLS technique is illustrated by the difference between LLS based controller responses (thick grey line) and those of the optimal one, based on an perfect knowledge of the process (dots). On the contrary, the Simplex based controller (thick black line) produces an almost overlapping response with respect to the optimal one. (Relative values of $DITAE$ are 1.2 vs. 0.05).

In the second case (b), the error obtained by LLS is larger and leads to a very incorrect model; as a consequence, the response of the controller based on this model becomes unstable, once applied to the real process. The SPX controller still behaves very reasonably. (Relative values of $DITAE$ are 80 vs. 0.68).

Further details regarding the efficiency of the SPX technique can be highlighted by analyzing the effect of the ratio θ/τ and of the process structure. Processes have been classified in three categories: time delay dominant ($\theta \gg \tau$, namely $\theta/\tau = 5 \div 10$), lag dominant ($\theta \ll \tau$, namely $\theta/\tau = 0.1 \div 0.25$) and intermediate cases ($\theta \backsimeq \tau$).

All types of disturbance (equation Eq. 5-25) have been studied for each process family: minimum $DITAE$ values are obtained for the case of equal model and process structure. Average and maximum $DITAE$ values are reported in table 4.1.

Table 4.1: *Average (and maximum) values of $DITAE$ for different process dynamics, Simplex Method*

Process	$\theta \gg \tau$	$\theta \backsimeq \tau$	$\theta \ll \tau$
$FOPTD$	0.114 (0.61)	0.048 (0.27)	0.180 (0.67)
$SOPTD$	0.096 (0.47)	0.087 (0.120)	0.107 (0.71)
$HOPTD$	0.116 (0.52)	0.109 (0.48)	0.123 (0.75)

From table 4.1 it can be observed that:

- the $\theta \backsimeq \tau$ case gives lower $DITAE$ values of (both average and maximum).
- average $DITAE$ values are always small (less than 0.18), indicating very good identification and closed loop performance.
- the largest maximum value (0.75 for the case of lag dominant high order process) still indicates good performance.

4.5 Robustness of the procedure

4.5.1 Plant noise

Robustness tests to process noise were performed in simulation by adopting the $Simulink$ block *band limited white noise*. Good results were also shown in the presence of large noise to signal amplitude ratios ($\simeq 30\%$), certainly larger with respect to most industrial plants. More important is the application described in [7] on plant data to test its robustness in case of plant noise. The section of the plant under exam produces *SBR* (Styrene-Butadiene Rubber); it operates in continuous even though organized in campaigns, whose duration depends on the operativity of the plant where the section is included. Here the main reaction of polymerization is examined, a simplified scheme of the plant is depicted in figure 4.8.

The core of the process is the reactor where the polymerization takes place; it presents a jacket with a double purpose: in the transitory phase an heating fluid is processed to help the mass in the reactor to reach the ignition temperature; on the contrary, in steady state conditions a cooler fluid is feeded to maintain a stable temperature, compensating the reaction heat.

Temperature of the bulk is the most important variable of the process because of its influence on the quality of final product. Reagents are preheated properly and a temperature controller is implemented; its importance is highlighted by the usage of a derivative action, while all other controller only have proportional-integral components. On the reactor a cascade scheme is adopted again to maintain a constant

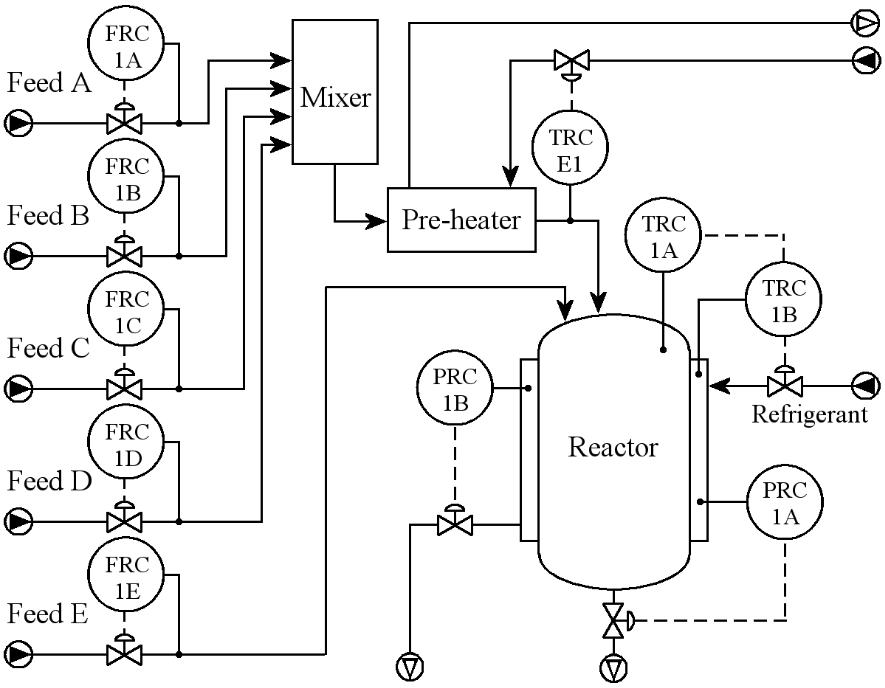


Figure 4.8: *Simplified scheme of the SBR plant*

temperature: the external loop controls the reaction temperature and generates the set point to the internal loop, acting on the basis of the jacket temperature. This choice is due to the strict tolerance allowed on temperature of the bulk ($\pm 3°C$): a disturbance in jacket temperature can be counteracted by internal loop rapidly, even before its effect would affect the external loop.

Furthermore the reactor surface is affected by a fouling phenomenon whose effect in time is the increasing in circulation of the cold fluid entering in the jacket to maintain constant the heat transfer.

Data collected in two consecutive days demonstrate that the plant is not heavily subjected to disturbances; however in the reactor zone two significant oscillations has been detected. Collected data and simulated responses obtained through the application of the simplex method are depicted for both disturbances in figure 4.9; the convergence of the simplex on a perfect fit is evident (gray line).

The gray curves depicted in figure 4.9 are generated by the two models of process (P) and disturbance (P_d) described in equation Eq. 5-26 for the first case and Eq. 5-27 for the second case.

$$P = \frac{0.46 \cdot e^{-10.7s}}{139.2s^2 + 73.6s + 1}; \quad P_d = \frac{1}{30.4s^2 + 4.2s + 1} \qquad \text{(Eq. 5-26)}$$

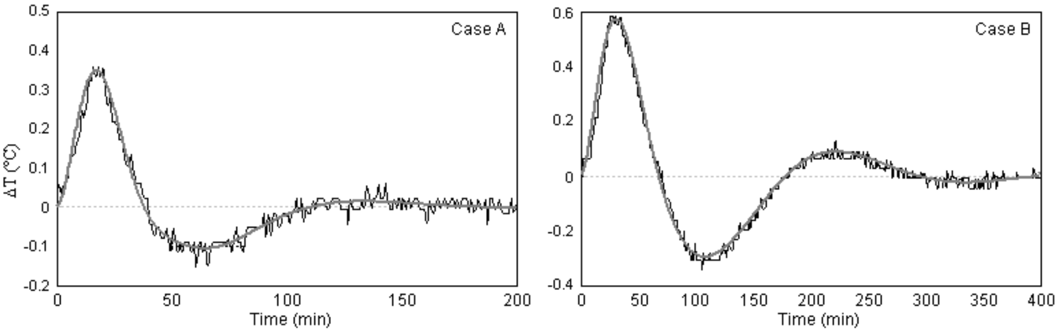


Figure 4.9: *Application of the simplex method on two disturbances: collected data in black; simplex method response in gray*

$$P = \frac{0.56 \cdot e^{-1.8s}}{1641.87s^2 + 166.94s + 1}; \quad P_d = \frac{1}{127.91s^2 + 13.34s + 1} \qquad \text{(Eq. 5-27)}$$

The minimum $ITAE$ controller, designed on the models, could bring to a faster disturbance rejection; a forecast of the obtainable performance is depicted in figure 4.10 for the two cases.

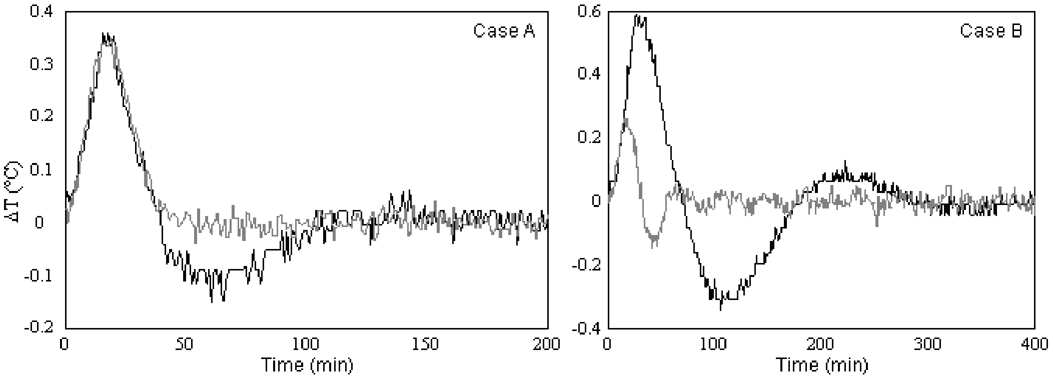


Figure 4.10: *Application of the minimum $ITAE$ controller based on models with addition of simulated noise: in black collected data, in gray obtainable performance.*

The second case shows a large improvement due to a different tuning; however in case of small time delay (equation Eq. 5-27) the minimum $ITAE$ controller may bring to small gain and phase margins (see [44]), thus generating large sensitivity to process uncertainty. Henceforward, a more detailed analysis on this issue has been accomplished as described in the following section.

4.5.2 Robustness to process uncertainty

A more important issue is robustness to process uncertainty: it is evident that inaccurate models can generate controllers with poor robustness, whatever the tuning rule or design technique. Nevertheless, even a perfect identification can cause a low robustness if the controller is too aggressive, according to the adopted design/tuning technique. Owing to their low order structure, the PI controllers adopted in our case, can not be pushed to optimal performance and therefore have some intrinsic robustness to model error, but the problem is still present. Estimation of process uncertainty from collected data is not easy to perform. It is much simpler to evaluate gain (G_M) and phase (P_M) margins of the controller designed on the identified model and, if low values are found, improve robustness by changing tuning or design technique.

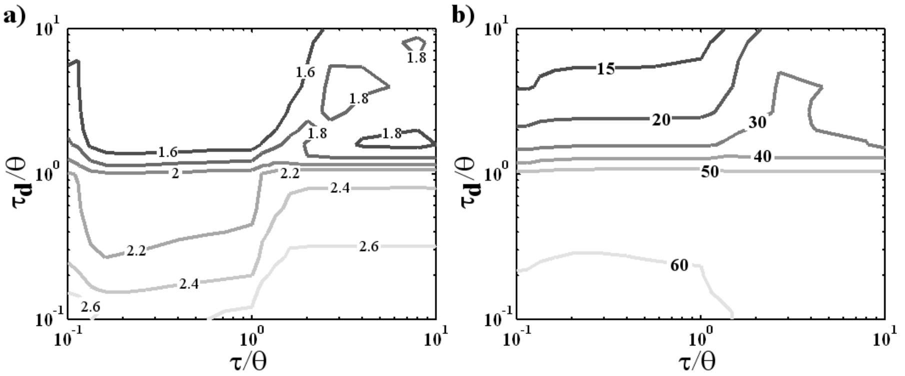


Figure 4.11: Variations of Gain Margin (a) and Phase Margin (b) for controllers designed for minimum $ITAE$, as a function of τ/θ and τ_d/θ; (log scales).

This point has been intensively evaluated in [44] and [45], for different controller structures and design specifications. Among other things, it is highlighted that controllers designed on the basis of minimizing an error function can be sensitive to robustness problems; in particular, the minimum $ITAE$ controller is more robust when designed for set-point tracking, while lower values of G_M and P_M are obtained for disturbance rejection. An evaluation of G_M and P_M has been performed for the case of PI controllers designed for disturbance rejection across the entire range of P and P_d dynamics (figure 4.11): it has been confirmed that low G_M and P_M values can be obtained for the case of lag dominant disturbances ($\tau_d > \theta$). In the case of set-point tracking, where the disturbance dynamic is not important, G_M and P_M have been analyzed as a function of τ/θ: it is confirmed that also in this case, the controller may show poor robustness, for lag dominant processes ($\tau > \theta$).

As a conclusion from the previous analysis, robustness must be explicitly taken into account when designing the controller, and this can be done by introducing a constraint on acceptable gain and phase margin values. The objective function is modified as:

$$\underset{C}{min} \int |e(t)| \cdot t \cdot dt \quad s.t. \left\{ \begin{array}{l} G_M > \bar{G}_M \\ P_M > \bar{P}_M \end{array} \right. \qquad \text{(Eq. 5-28)}$$

where $\bar{G}_M$ and $\bar{P}_M$ are the gain and phase margin boundaries respectively, which can be defined by the operator depending on knowledge of the plant and on the desired response. Implications of robustness constraints on controller design is illustrated in figure 4.12: when one of the two constraints becomes active, controller parameters are selected to respect the desired robustness margin.

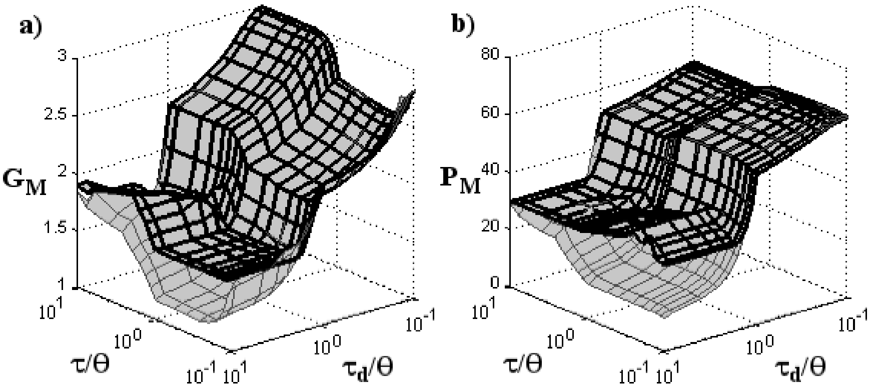


Figure 4.12: *Implications of constraints on G_m (a: $\bar{G}_M = 1.6$) and on P_m (b: $\bar{P}_M = 30$) in the design of minimum $ITAE$ controllers, as a function of τ/θ and τ_d/θ (log scales); gray layer: without; black mesh: with.*

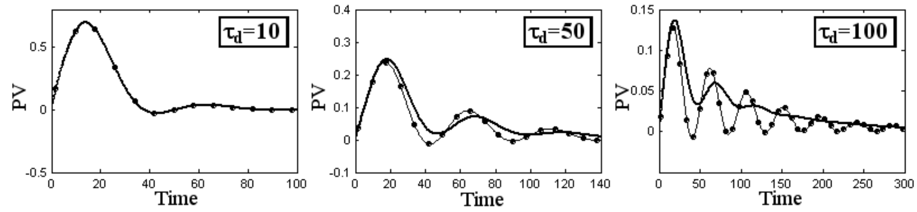


Figure 4.13: *Disturbance rejection for minimum $ITAE$ controllers with constraints on G_M and P_M (thick line) and without constraints(thin line with dots).*

The effect of this modification on closed loop performance, for the disturbance rejection case, is only shown when the original controller has poor robustness and therefore the constraint becomes active, i.e. in the case of lag dominant disturbance. Some significant results are presented in figure 4.13, for the case: $\tau = 10$, $\tau_d = 10, 50, 100$; Gain and Phase Margins are respectively equal to: $G_M =$

$2.09, 1.37, 1.17, P_M = 54.7, 22.4, 12.2$, for unconstrained controllers and to: $G_M > 2; P_M > 30$, for constrained controllers.

It can be seen that the respect of constraints has no effect at all on the response in many cases. For the case of large τ_d, the $ITAE$ optimal response given by the unconstrained controller, is faster but strongly oscillating, while respect of constraints generates a slower, but practically more desirable, response (even though it remains suboptimal according to the $ITAE$ criterion). Therefore respect of robustness criteria does not seem to negatively affect the proposed identification and design procedure.

Chapter 5

Design of a software package

For each of the studied techniques a relevant issues was about their application on real plants and their usage by people which will not be interested in understanding background theories. It has to be developed a friendly interface software organized in a module sequence easy to be implemented with few simple instructions. The main purpose of this software, named PCU, acronym for Plant Check Up, is to give operators a standard procedure for the application of studied techniques to classify loops behavior and to discover sources of low performance. The software can be applied either on plant data sets or simulated data sets; for this reason it can be used also as demonstrative or educational tool.

5.1 General overview of PCU package

To develop this software the MATLAB programming language was adopted for several reasons. Principally it has some advantage with respect to other *high level* programming languages as well as Visual Basic and Visual C++ which do not have built-in functions, whose application simplify calculations. For instance the Simplex method, implemented with few modifications to the MATLAB function *fminsearch.m*, would require a complete sequence of commands in new subroutines with other languages. Furthermore, from 5.3 version, MATLAB gives the possibility of using a Graphical User Interface (GUI) with all the standard graphical controls, covering all advantages Visual Basic and Visual C++ had, thanks to their easy procedure to build friendly interface programs. Simulink additional components are not used, substituting them with faster procedures, not based on differential equations (as Simulink does) but developed in discrete domain to obtain, as consequence, a lower calculation time.

Certainly it must be considered that a *high level* programming language has

[*] Portions of this chapter are published in [46] [47] and its extended version [35]

a main drawback in its complexity and as consequence applicability on DCS or PLC, where used.

Simplifying, internal DCS and PLC structures are composed by a motherboard connected directly with alimentation, CPU and I/O (Input/Output) slots, connected with other I/O boards which dialogue directly with the plant, as shown in figure 5.1.

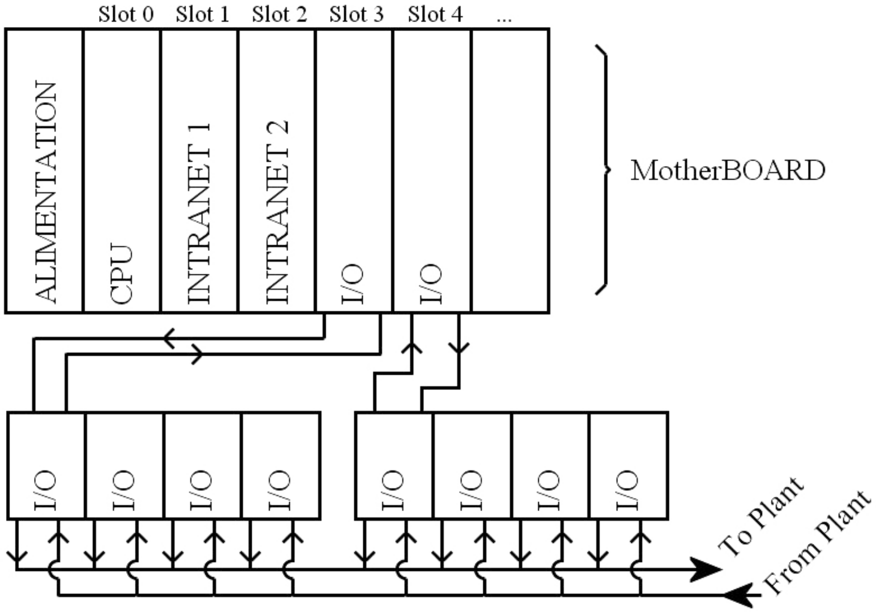


Figure 5.1: *Motherboard scheme and I/O board connections*

With this structure, and considering that for instance $PLCs$ must be built and programmed as *IEC1131-1* normative describes, only a simplified *high level* language, very similar to a simple PASCAL, can be adopted. The programming language, called structured text, can only assess: bit/words assignments, IF-THEN-ELSE cycles, WHILE-REPEAT cycles etc.. Obviously MATLAB built-in functions can not be used directly with this structure and only an off-line application can be performed. For on-line procedures the software must be translated in commands of a lower level language. The off-line application has its advantages: a general overview of all considered control loops performances can be done and, repeating the procedure, the status of the plant can be monitored in time. Of course an on-line application allows the operator to intervene immediately whenever a low performance source is discovered. The system described in the sequel is designed for an off-line application, further developments will be done in future to adapt it to on-line application.

The software scheme is described in figure 5.2.

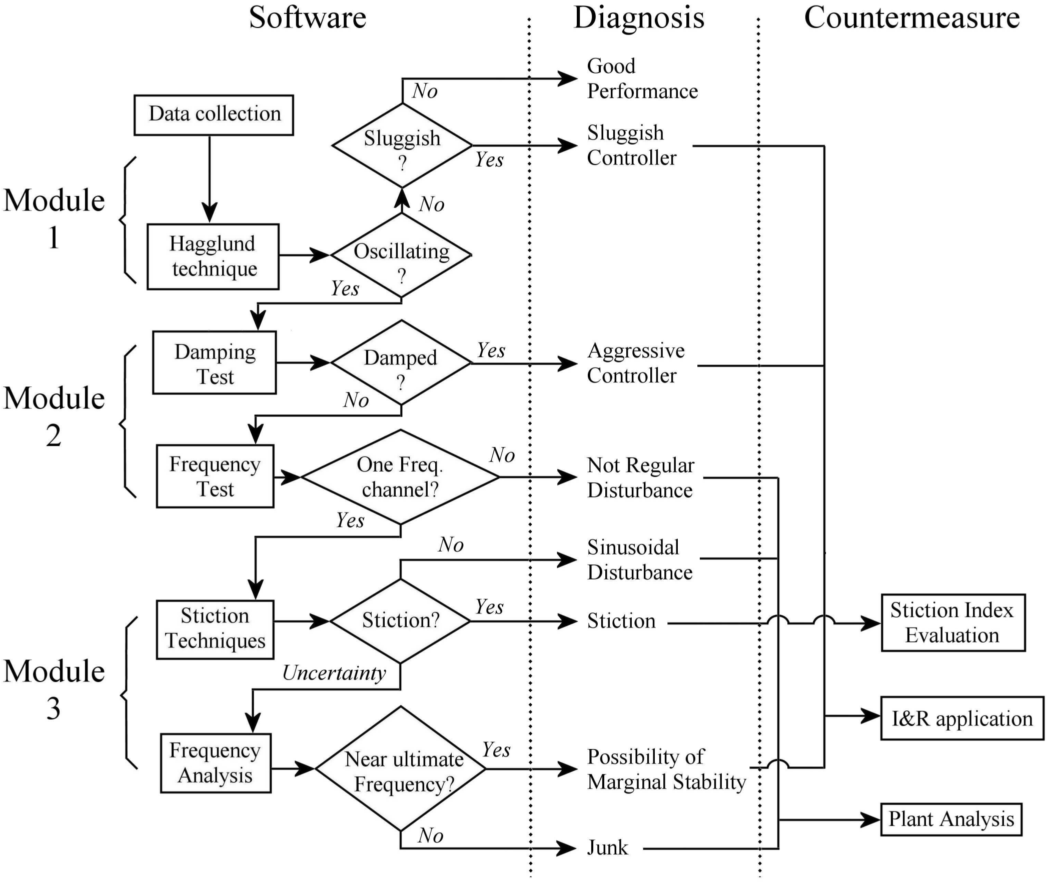


Figure 5.2: *General scheme of the system divided by modules organization*

All packages and tools will be described more in detail in this chapter; here a brief summary is reported. The system can be divided in three parts to be implemented in sequence:

- **Data acquisition:** this section is to be used to import data directly in a format recognizable by MATLAB from Excel files, chosen as templates because of their wide diffusion.
- **Diagnosis of loops performances:** this section can be subdivided into three modules: the first of them is used to exclude from analysis good performing loops, the second to find oscillating loops and the third to detect stiction.
- **Report and countermeasure:** at this point of the application of the software, loops are completely classified and a report is produced. If the cause of low

performance is due to controller tuning a *I&R* (Identification and Retuning) tool can be applied. If the problem resides in a stuck actuator, a stiction index can be evaluated. For other cases a more detailed analysis must be done both on control loop and plant units.

Only main file named *pcu.m* is shown directly into the main folder of the program; all other files are divided in three subfolders: *loopanalysis subfolder* which contains files relative to the application of Module 1 and Module 2; *stiction subfolder* with files relative to the application of Module 3 and finally *controller subfolder* with files the system needs to implement $I\&R$ tool. To launch the program the user only needs to select the folder which contains the *pcu.m* file as *Current Directory* in MATLAB and type *pcu* in the *Command window*. After this, all operations can be done by following indications the system provides through graphical windows, as explained more in detail in next sections.

5.2 Data acquisition and management

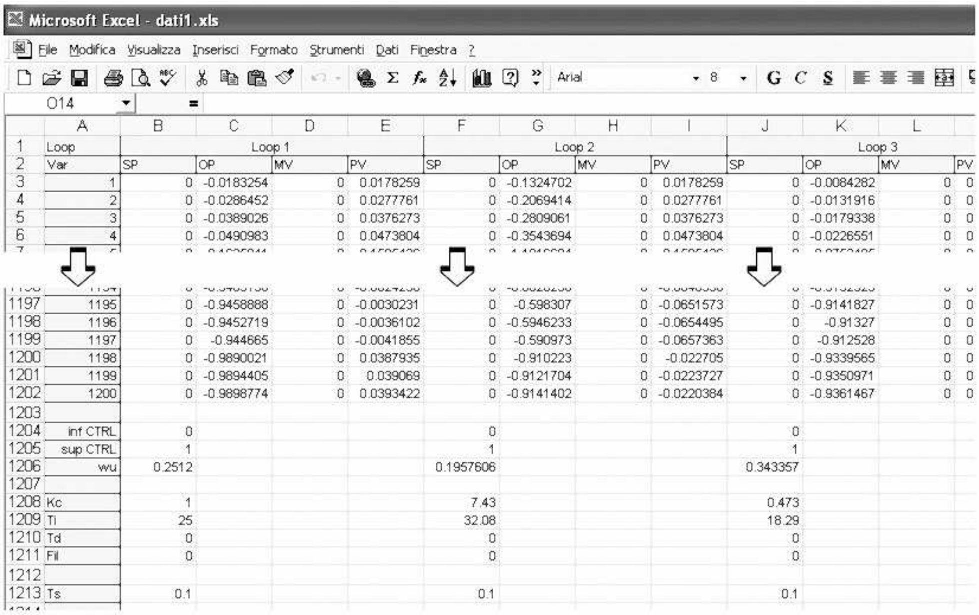
The program starts importing in MATLAB collected data sets for off-line analysis: this operation must be done correctly to avoid errors in application of the software. In particular data sets must be well organized in a fixed structure, defined in a Excel template.

For this reason, initially a detailed description of simple rules for the organization of the Excel file will be reported and an example of the template to be used will be provided. The second step of the procedure is to activate the main window of the software (called *Navigation Window*): the button to acquire data is the only one enabled, forcing users to open the *Data Acquisition* window. If Excel file is well organized, MATLAB can handle all imported data building vectors and matrixes, which will be recalled by other files; otherwise an error message will be shown and the system will return to the Navigation Window. Users only need to follow procedures as shown by the graphical interface, no parameters must be defined in this section.

5.2.1 Excel file template for data-sets

Before starting the procedure, data sets must be properly stored in an Excel file with all information about loops, required by the system for all analysis. In particular set point (SP), controlled variable (PV) and control output (OP) values must be collected; when possible also manipulated variable (MV) should be imported in the Excel template. Other variables the system needs are: control range; ultimate frequency (or its approximation evaluated as explained by Hägglund [1]); control parameters as well as gain, integral time and, in case of PID structures, derivative time and filtering constant; the sampling time value.

All information must be stored in a fixed scheme equal to the one shown in figure 5.3:



	A	B	C	D	E	F	G	H	I	J	K	L	
1	Loop	Loop 1				Loop 2				Loop 3			
2	Var	SP	OP	MV	PV	SP	OP	MV	PV	SP	OP	MV	PV
3	1	0	-0.0183254	0	0.0178259	0	-0.1324702	0	0.0178259	0	-0.0084282	0	0
4	2	0	-0.0286452	0	0.0277761	0	-0.2069414	0	0.0277761	0	-0.0131916	0	0
5	3	0	-0.0389026	0	0.0376273	0	-0.2809061	0	0.0376273	0	-0.0179338	0	0
6	4	0	-0.0490983	0	0.0473804	0	-0.3543694	0	0.0473804	0	-0.0226551	0	0
1197	1195	0	-0.9458888	0	-0.0030231	0	-0.598307	0	-0.0651573	0	-0.9141827	0	0
1198	1196	0	-0.9452719	0	-0.0036102	0	-0.5946233	0	-0.0654495	0	-0.91327	0	0
1199	1197	0	-0.944665	0	-0.0041855	0	-0.590973	0	-0.0657363	0	-0.912528	0	0
1200	1198	0	-0.9890021	0	0.0387935	0	-0.910223	0	-0.022705	0	-0.9339565	0	0
1201	1199	0	-0.9894405	0	0.039069	0	-0.9121704	0	-0.0223727	0	-0.9350971	0	0
1202	1200	0	-0.9898774	0	0.0393422	0	-0.9141402	0	-0.0220384	0	-0.9361467	0	0
1203													
1204	inf CTRL	0				0				0			
1205	sup CTRL	1				1				1			
1206	wu	0.2512				0.1957606				0.343357			
1207													
1208	Kc	1				7.43				0.473			
1209	Ti	25				32.08				18.29			
1210	Td	0				0				0			
1211	Fil	0				0				0			
1212													
1213	Ts	0.1				0.1				0.1			

Figure 5.3: *Example of the Excel Template where data sets are stored*

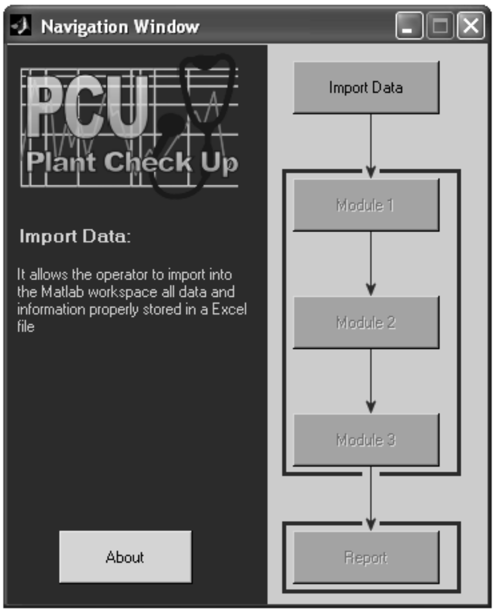
As general rule, gray columns are fixed and should not be modified, unless users want to adapt the sequence of numbers in the first column to the amount of collected data of the single variable. Four columns are dedicated to each loop, containing in a fixed sequence set point, controller output, manipulated variable and controlled variable. If MV in unknown a series of zeros should be written on its relative column.

Below data sets, a portion of the Excel template is dedicated to all the other variables: $infCTRL$ and $supCTRL$ represent labels respectively for lower and upper limits of the control range; wu is the ultimate frequency of the system; Kc, Ti, Td, Fil are controller parameters (gain, integral time, derivative time and filtering constant) and Ts the sampling time. If wu is unknown, users can choose between two options: either using its estimation proposed by Hägglund [1] or leaving empty the relative cell.

Obviously this template can not be adapted easily to on-line application of the software, a further step in a development of an industrial tool will be done to generate a more versatile interface to collect data.

5.2.2 Data Acquisition Window

Once the Excel file is ready, users can launch the software by typing in the MATLAB command window the word *pcu*. The Navigation Window will appear as show in figure 5.4:



Import Data:

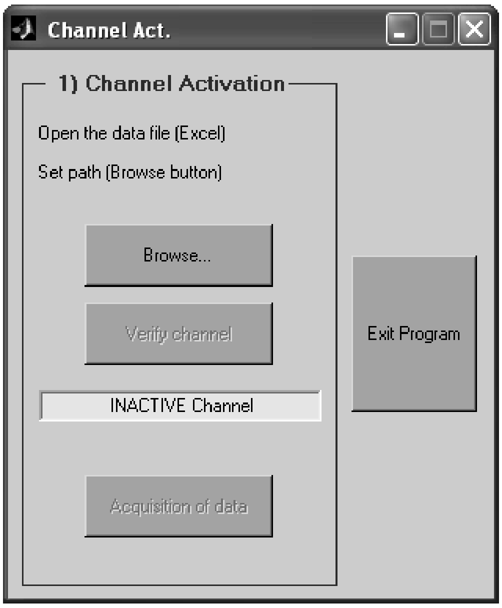
It allows the operator to import into the Matlab workspace all data and information properly stored in a Excel file

Figure 5.4: *Navigation Window*

The only possible choice is to press the *Import Data* button, thus selecting the Channel Activation window, whose main purpose is to open the channel between MATLAB and Excel to import data (figure 5.6).

This graphical window shows a brief summary of the operation the user needs to follow. The first step allows the system to open the Excel file containing data from plant; then, pressing the *Browse* button, the file path can be defined directly as shown in figure 5.6.

Once the file is selected the *Verify Channel* button in Channel Activation Window is enabled: by pressing it the software compares names between the opened Excel file and the selected one in the Browse window. If a correspondence is not found the system shows a red string (*Inactive channel*) and does not allow the user to proceed to next steps. Otherwise, the channel is activated and data sets can be imported in MATLAB environment by pressing the *Acquistion Data* button. At the end of this operation the system returns to the Navigation window activating the *Module 1* button.

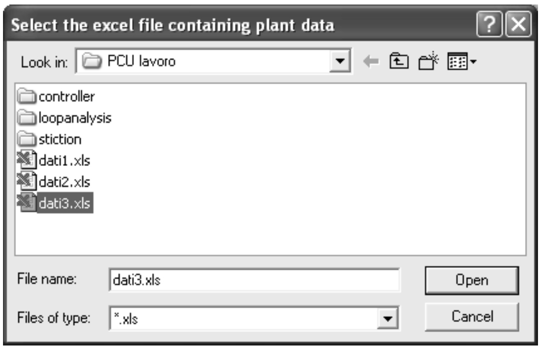


Open the data file (Excel)

Set path (Browse button)

INACTIVE Channel

Figure 5.5: *Channel Activation window: used to import data in MATLAB environment*



Look in: PCU lavoro

File name: dati3.xls

Files of type: *.xls

Figure 5.6: *Browse window which allows users to select Excel files containing data sets*

5.3 Module 1: loop performance evaluation

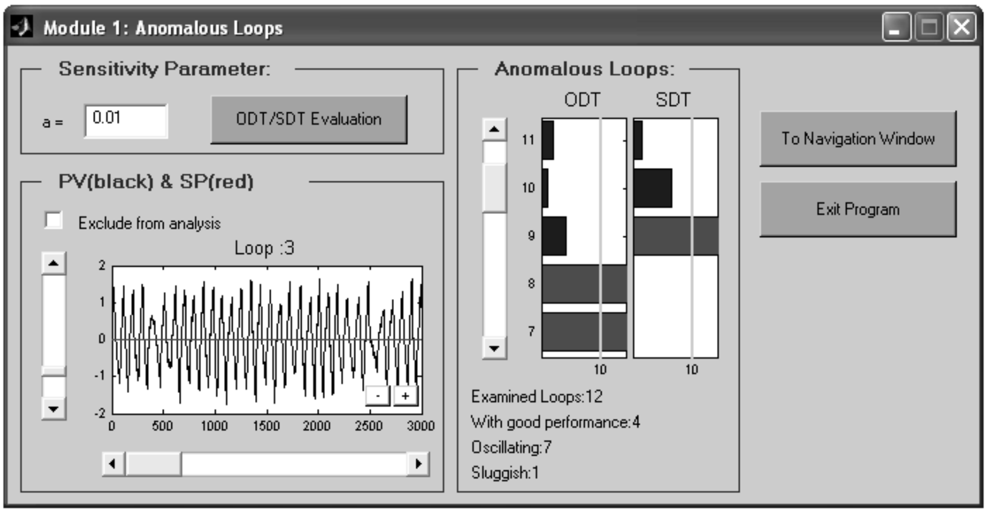
This module receives data sets imported by Excel files and all information about control loops as input. The output is a first classification of loops labelling them as:

- **Good performance**: loops which are affected only by plant noise and consequently do not require further analysis
- **Oscillating**: loops with a sequence of more than 10 peaks in IAE_{Mod}, as requested by Hägglund technique [1] (see chapter 2).

- **Sluggish**: loops with significant errors but without a sequence of peaks.

.

Module 1 is activated by pressing the relative button in the navigation window and the window depicted in figure 5.7 is opened.



a = 0.01

□ Exclude from analysis

Examined Loops:12
With good performance:4
Oscillating:7
Sluggish:1

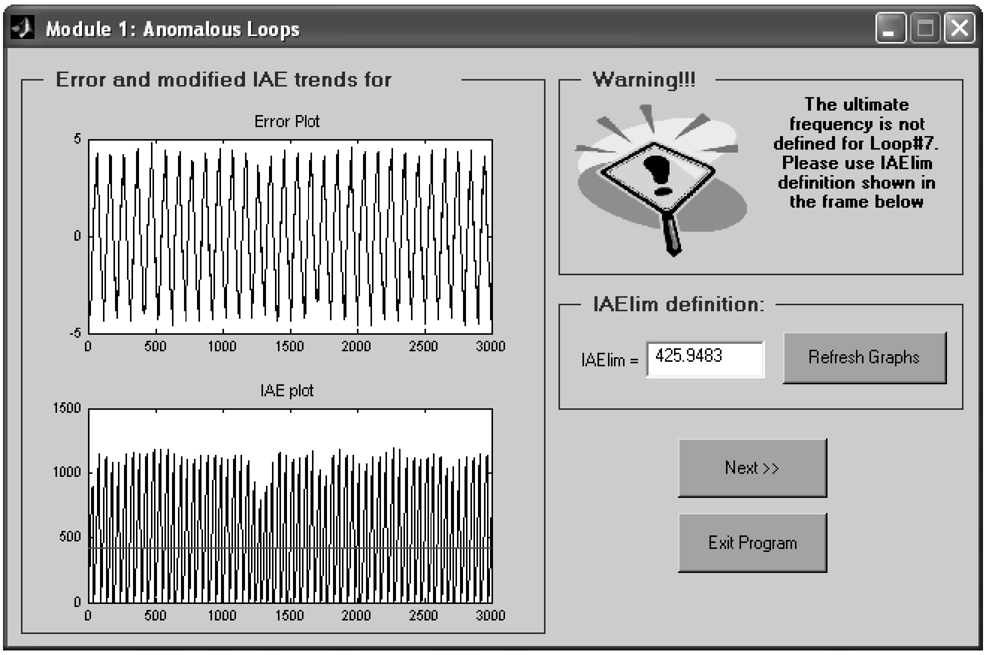
Figure 5.7: Module 1: Anomalous Loops window

Users only need to insert a proper value for Hägglund parameter *a* as described in chapter 2 (default value is chosen as 0.01) and press the *ODT/SDT Evaluation* button, where ODT means Oscillation Detection Technique and SDT Sluggishness Detection Technique.

In the lower section a graph is reported, where PV and SP values for all loops are depicted: a vertical scroll bar allows the user to change loop; an horizontal scroll bar moves abscissa, zoomed by + and - buttons. If the user wants to exclude some loops because of a wrong collection of data (system failures, shutdown or start-up of the plant etc.), the box labelled *Exclude from analysis* can be selected.

In the section labelled *Anomalous Loop*, two bar graphs are shown; the first one relative to the ODT index, the second one to the SDT index: when bars overcome the gray line an anomaly is detected; otherwise the loop under exam is considered with good performance. Under graphs a short summary of the result of the application of this module is reported.

If, during the analysis, for some loops the ultimate frequency was not known and not written in the Excel file, the software opens for each of them a window (figure 5.8) in which the error $SP - PV$ and IAE_{Mod} peaks from Hägglund technique [1] are depicted.



The ultimate frequency is not defined for Loop#7. Please use IAElim definition shown in the frame below

IAElim = 425.9483

Figure 5.8: $IAElim$ *selection in case of unknown* ω_U

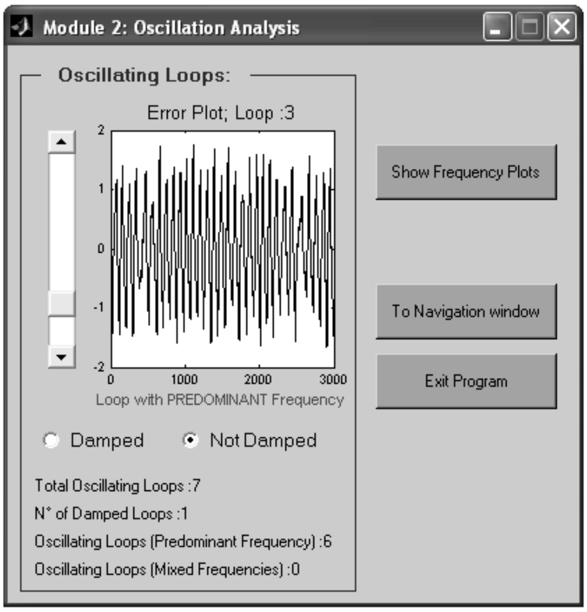
The user can choose a value for IAE_{Lim} varying it on the basis of their experience to set a good threshold value to establish the level of significance of oscillations; a graphical confirmation can be obtained by refreshing graphs with the relative button.

5.4 Module 2: properly oscillating loops

Once Module 1 has been processed, the system enables the Module 2 button on the Navigation Window (figure 5.4), which allows the user to perform time domain and frequency domain analysis on loops tagged oscillating by module 1. The result is a further classification: damped oscillations (due to aggressive controllers); oscillations characterized by a predominant frequency (due to different causes to be detected in next steps of the procedure); oscillations due to the combination of different frequency channels (not regular disturbances).

Module 2 is activated similarly to Module 1 by pressing the relative button in the navigation window, thus opening the window depicted in figure 5.9.

Users have not to make operations in this module; application of time and frequency domain techniques is completely automatic as well as indications on damping and frequency channels of the oscillation. However results can be modified on the basis of operator knowledge; hence the system can be properly forced selecting



○ Damped ● Not Damped

Total Oscillating Loops :7

N° of Damped Loops :1

Oscillating Loops (Predominant Frequency) :6

Oscillating Loops (Mixed Frequencies) :0

Figure 5.9: *Module 2: Oscillation Analysis*

the option field named *Damped*, thus choosing the application of the $I\&R$ tool on loop under exam, or *Not Damped*, thus excluding controller tuning as source of poor performance. Similarly to Module 1 graphical window, a vertical scroll bar can be used to select loops and a brief summary of Module 2 results is reported.

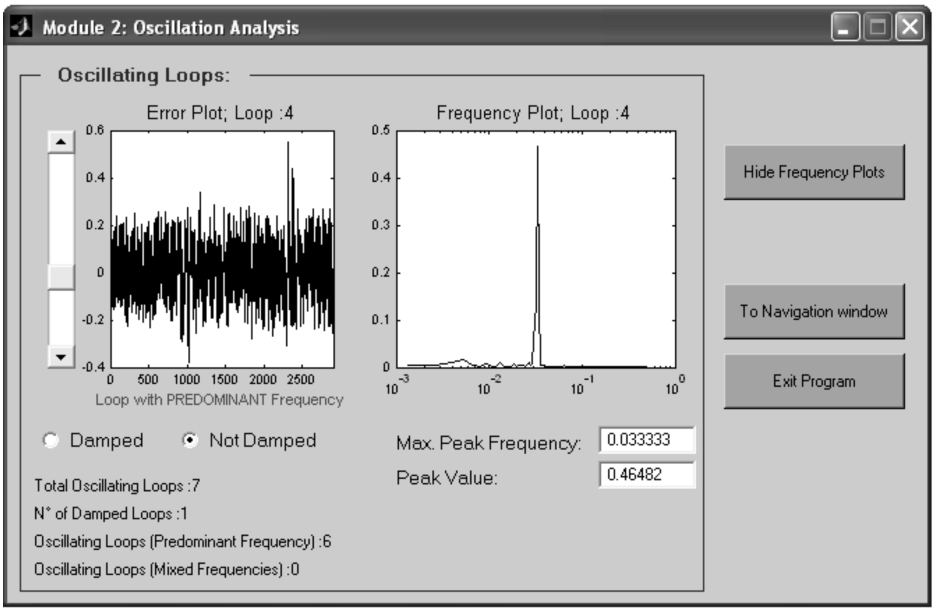
By pressing the button named *Show Frequency Plots*, Module 2 window is enlarged showing also information on frequency channels of the signal (figure 5.10). Power Spectra of signals are shown in the *Frequency Plot* graph, whose abscissa is represented in a log scale to highlight also low frequency peaks. Fields under graph report maximum value of the Power Spectrum and its corresponding frequency.

5.5 Module 3: stiction detection tools

Module 3 can be applied over loops classified as oscillating with a predominant frequency to distinguish among stiction on actuators, sinusoidal disturbances or marginal stability conditions. Module 3 button on Navigation window will open the relative graphical window (reported in figure 5.11).

A plot of the error ($SP - PV$) is again shown together with the vertical scroll bar usable to select the loop to be analyzed.

Buttons in the lower section allows the user to choose a technique among MV versus OP comparison, Cross Correlation, Bichoerence, Relay Technique. For each of them a graphical verification is obtained as shown for instance in figure 5.12 for



○ Damped ● Not Damped

Max. Peak Frequency: 0.033333

Peak Value: 0.46482

Total Oscillating Loops :7

N° of Damped Loops :1

Oscillating Loops (Predominant Frequency) :6

Oscillating Loops (Mixed Frequencies) :0

Figure 5.10: *Module 2: Oscillation Analysis with frequency analysis*

a case of stiction.

Under graphs, indexes relative to the chosen technique are reported; for MV versus OP comparison (figure 5.12a), a warning is shown advising users that PV substitutes MV when the latter variable is not known. In relay technique plot (figure 5.12d) all approximating curves are depicted and the vertical scroll bar can be used to look at different half-cycles of the oscillation. A summarizing string (either *Stiction* or *Disturbance* or *Uncertainty*) is displayed in gray fields over graph (figure 5.11) as results of the application of each technique.

A control valve is automatically labelled sticky when at least two of the listed techniques indicate it (filling the *Stiction* check box, otherwise empty). However, this box can be selected by the user to force the system in associating loop under exam to a stiction problem and to evaluate a Stiction Index in next steps.

Users can choose to return to Module 2 and modify results by forcing the software as explained in previous section, or to proceed to the final report. In the latter case, if some loops have not been examined by Module 3, the system produces a warning as shown in figure 5.13.

5.6 Report and countermeasures

Application of the three Modules in sequence allows the user to diagnose completely the cause of poor performance for all considered loops. A report can be

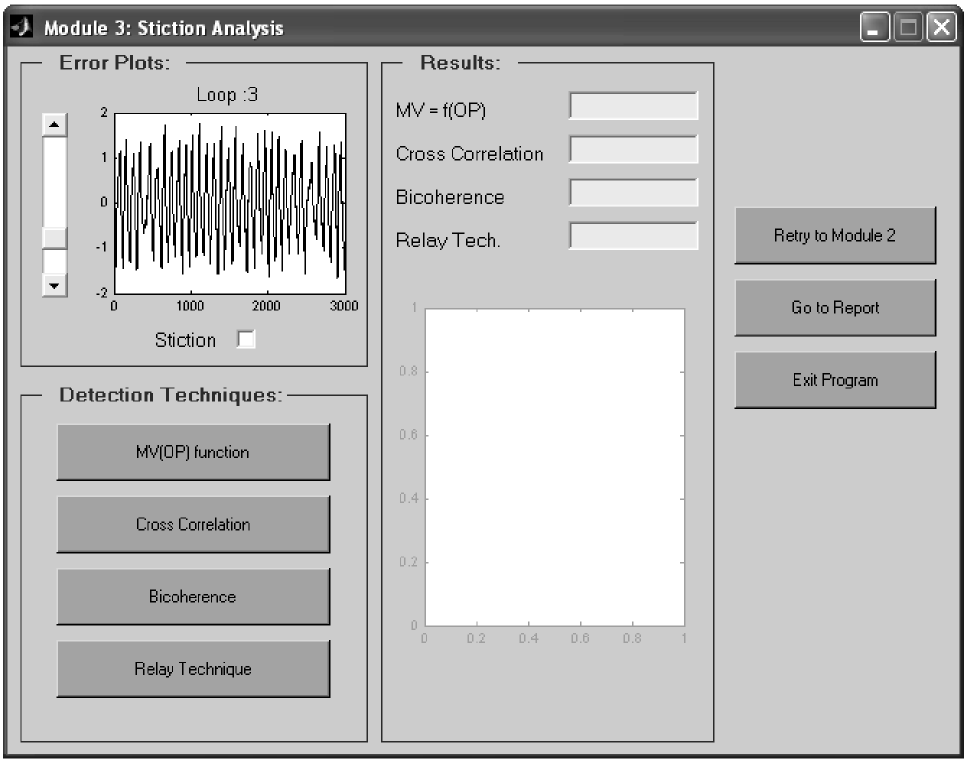
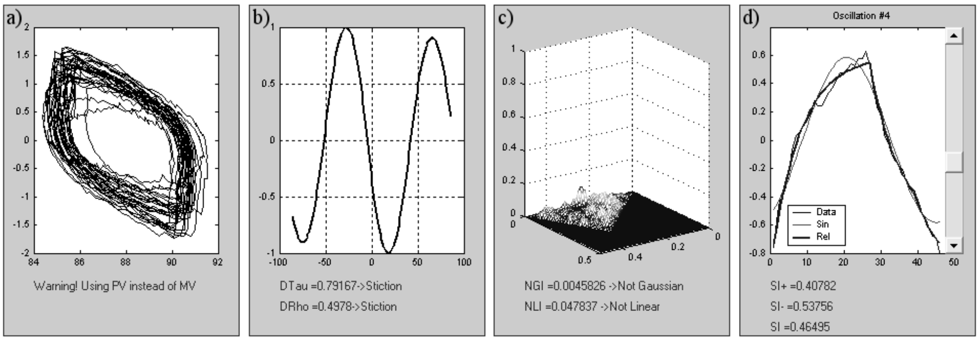


Figure 5.11: *Module 3: Stiction Analysis*



Warning! Using PV instead of MV

DTau =0.79167->Stiction
DRho =0.4978->Stiction

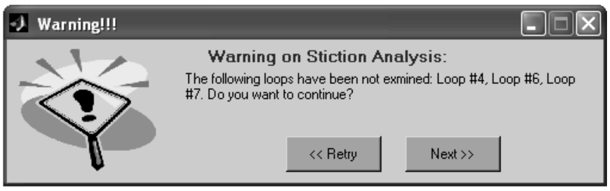
NGI =0.0045826 ->Not Gaussian
NLI =0.047837 ->Not Linear

SI+ =0.40782
SI- =0.53756
SI =0.46495

Figure 5.12: *Examples of results of stiction techniques: a)MV versus OP comparison, b) Cross Correlation, c) Bichoerence, d) Relay Technique.*

shown by pressing relative button on Navigation Window, which opens a graphical window similar to the one depicted in figure 5.14.

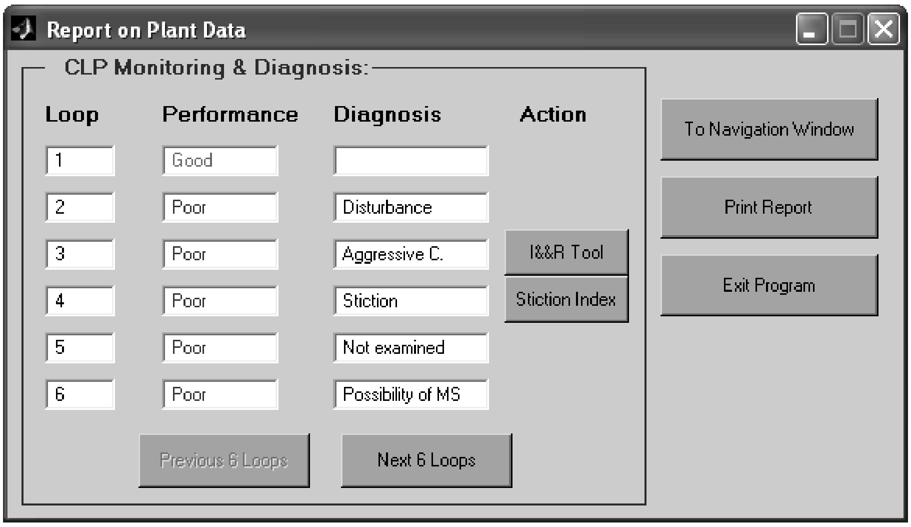
To facilitate the reading of results by users, loops are listed in packets of six selected by *Previous 6 loops*/*Next 6 loops* buttons. Excluding the column containing loop numbers, only two columns are filled: the first quotes information about the monitoring phase (good/poor performance); the second about the diagnosis phase.



Warning on Stiction Analysis:

The following loops have been not exmined: Loop #4, Loop #6, Loop #7. Do you want to continue?

Figure 5.13: *Example of warning message for not examined loop*



CLP Monitoring & Diagnosis:

Loop	Performance	Diagnosis	Action
1	Good		
2	Poor	Disturbance	
3	Poor	Aggressive C.	I&&R Tool
4	Poor	Stiction	Stiction Index
5	Poor	Not examined	
6	Poor	Possibility of MS	

Figure 5.14: *Examples of Report Window*

The system can choose among nine labels to be used to fill the diagnosis column:

- **Excluded**: for loops that users do not want to analyze selecting the checking box *Exclude from Analysis* in figure 5.7.
- **Sluggish C.**: controller is acting too slowly, large errors are detected by SDT software through the application of Module 1 (figure 5.7).
- **Aggressive C.**: an aggressive tuning is adopted for this loop, detected by ODT software in Module 1 (figure 5.7) and damping analysis in Module 2 (figure 5.9).
- **Not Regular Disturbance**: detected in oscillating loops whose frequency content is composed by a mixture of different signals not prevailing on each other (Module 2, figure 5.9).
- **Stiction**: oscillating loops (Module 1) with a predominating frequency (Module 2) for which a problem on actuator is revealed by the application of several techniques listed in figure 5.11 (Module 3).

- **Disturbance**: for loops processed as the previous case but with clear evidence of a sinusoid oscillation or absence of non-linearities.

- **Possibility of MS**: for loops not belonging either to stiction or to disturbance cases, however showing a frequency near ω_U or its estimation.

- **Not Examined**: tag used for all loops, labelled as oscillating by Module 2, which are not processed by Module 3.

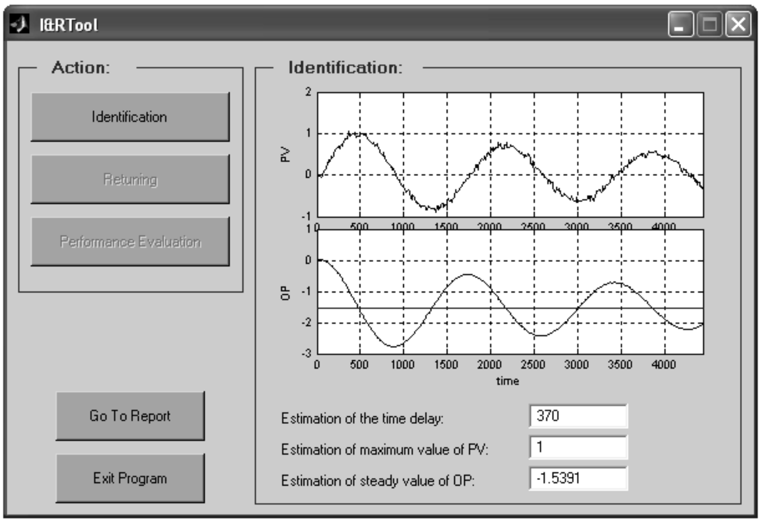
- **Junk**: every loop not belonging to one of the previous definitions goes into a Junk; for these cases a plant analysis or specific tests must be performed to diagnose the cause of poor performance.

When a new tuning can solve the problem (aggressive controller, sluggish controller, marginal stability conditions), on the right side of the diagnosis field an *I&R* button is depicted; by pressing it, the identification of a model and retuning of the controller can be performed. Similarly a *Stiction Index* button is shown for loops affected by stiction; for the other cases no buttons are displayed.

The *Print Report* button transforms the Report Window in a printable version and automatically opens the dialogue window with installed printers.

5.6.1 I&R tool

For loops, whose poor performance is due to controller tuning, the *I&R* button appears in Report Window (figure 5.14) and opens the *I&R Tool Window* shown in figure 5.15.



Estimation of the time delay: 370

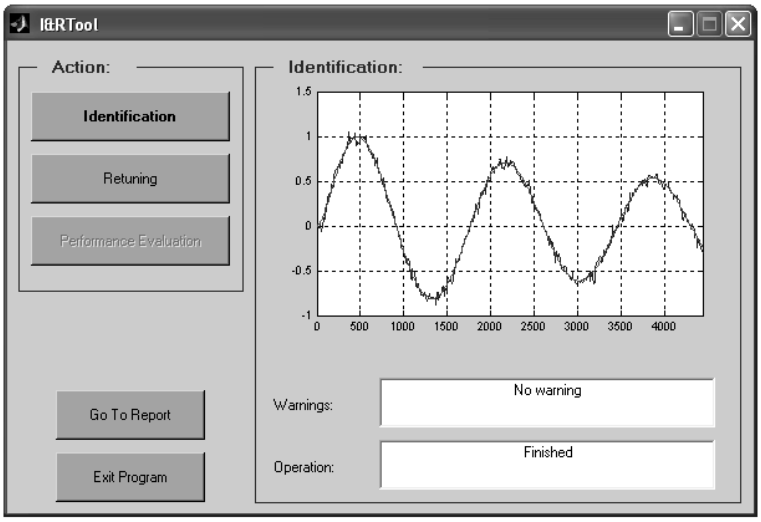
Estimation of maximum value of PV: 1

Estimation of steady value of OP: -1.5391

Figure 5.15: Examples of I&R Tool Window; identification phase

Firstly the system identifies a model of process and disturbance applying Simplex method as described in chapter 4. Automatically a first estimation of time delay, maximum value of PV and an estimation of OP value in stationary conditions are evaluated and shown under graphs; however the operator can also write different values in related fields using his plant knowledge.

Users can follow the proper sequence of operations by pressing buttons listed in the *Action* section and enabled only after the conclusion of the previous analysis. The *Identification* button starts the Simplex method procedure, which finishes displaying on the same graph collected PV data and evaluated PV response based on the model (example in figure 5.16).



Warnings: No warning

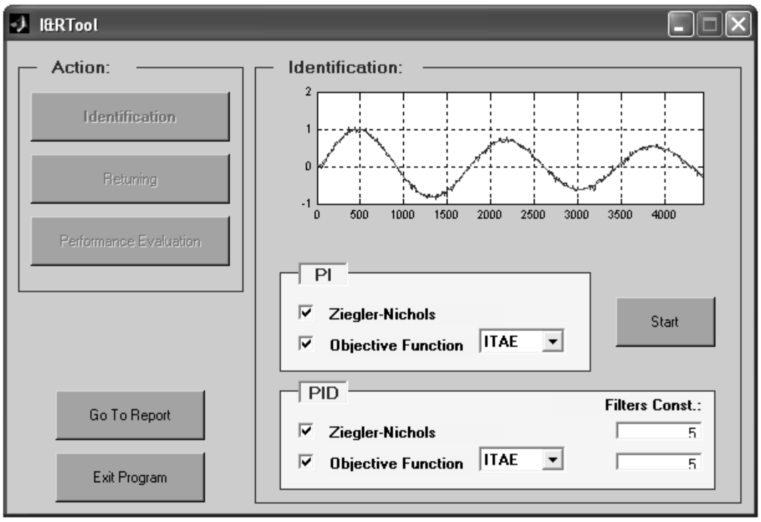
Operation: Finished

Figure 5.16: *Examples of I&R Tool Window; results of identification phase*

Simultaneously, because of availability of models for process and disturbance obtained in the previous step, *Retuning* button is enabled; retuning phase can start and figure 5.17 is shown.

Users can select techniques to be applied through check boxes, and for the case of *Objective Function*, both for PI and PID controllers, there is the possibility to choose the function error to be minimized in a popup menu composed by $ITAE$, ISE, IAE labels. To guarantee a proper transfer function for PID controllers, filter constant must be inserted in related fields by users. By default the system chooses a value equal to $T_i/10$, indicating with T_i the integral constant of the initially used controller. The *Start* button applies selected tuning techniques; however, as a first step, gain and phase margin limitations are requested:

By default a gain margin of 1.6dB and a phase margin of 30˙ are chosen. Users can change those values considering that controllers, obtained through unconstrained minimization of an error function, could be not robust when applied on real process



PI

☑ Ziegler-Nichols

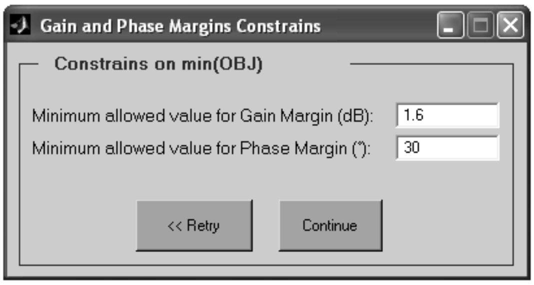
☑ Objective Function ITAE

PID

☑ Ziegler-Nichols 5

☑ Objective Function ITAE 5

Figure 5.17: Examples of I&R Tool Window; starting the retuning phase



Minimum allowed value for Gain Margin (dB): 1.6

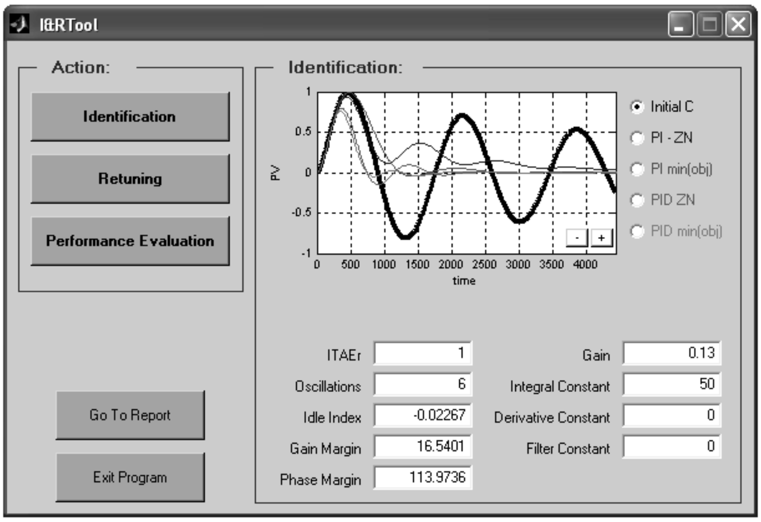
Minimum allowed value for Phase Margin (°): 30

Figure 5.18: Examples of I&R Tool Window; defining gain and phase margin

([45] and [44]).

When calculations for tuning techniques are accomplished, the *Performance* button in *I&R Tool* window (figure 5.15) is enabled: by pressing it the performance of all designed controllers is evaluated (an example is shown in figure 5.19).

Closed loop responses, displayed in the *I&R Tool* window in the performance evaluation phase, are obtained by using designed controllers in a feedback scheme with models of process and disturbance. Similarly to the error graph, depicted in figure 5.7, the $+/-$ buttons allow the user to zoom in and out. Check boxes on the right side of the graph, when pressed, highlight the response of the selected controller by increasing its thickness and contemporaneously all values listed below the graph are updated. In the first column, performance indexes are reported: $ITAEr$, to evaluate the improvement of the chosen controller with respect to the initially used (Rossi et al. [7]); $Oscillation$ and $IdleIndex$, to classify the behavior of the selected controller (respectively Hägglund [1], Hägglund [42]); *gain margin*



● Initial C
○ PI - ZN
○ PI min(obj)
○ PID ZN
○ PID min(obj)

ITAEr	1	Gain	0.13
Oscillations	6	Integral Constant	50
Idle Index	-0.02267	Derivative Constant	0
Gain Margin	16.5401	Filter Constant	0
Phase Margin	113.9736		

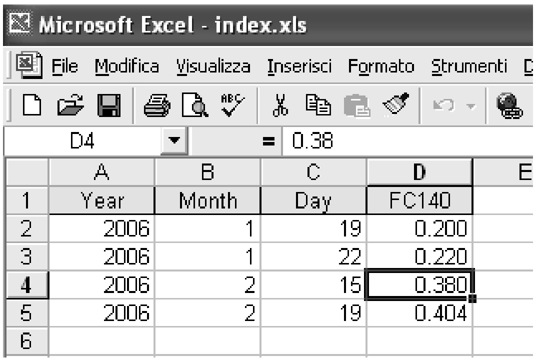
Figure 5.19: Examples of I&R Tool Window; performance evaluation

and *phase margin* (Ho et al. [45], Ho et al. [44]), to check robustness to process uncertainties. Second column only shows controller parameters (for PI controllers Derivative Constant and Filter Constant fields are filled with zero values).

After checking controller performance users can press the *Return to Report* button, thus having the possibility of examining other loops.

5.6.2 Stiction Index Tool

For loops to whom stiction is associated, there is the possibility to evaluate a Stiction Index. An Excel template is again provided for historical data of each loop:



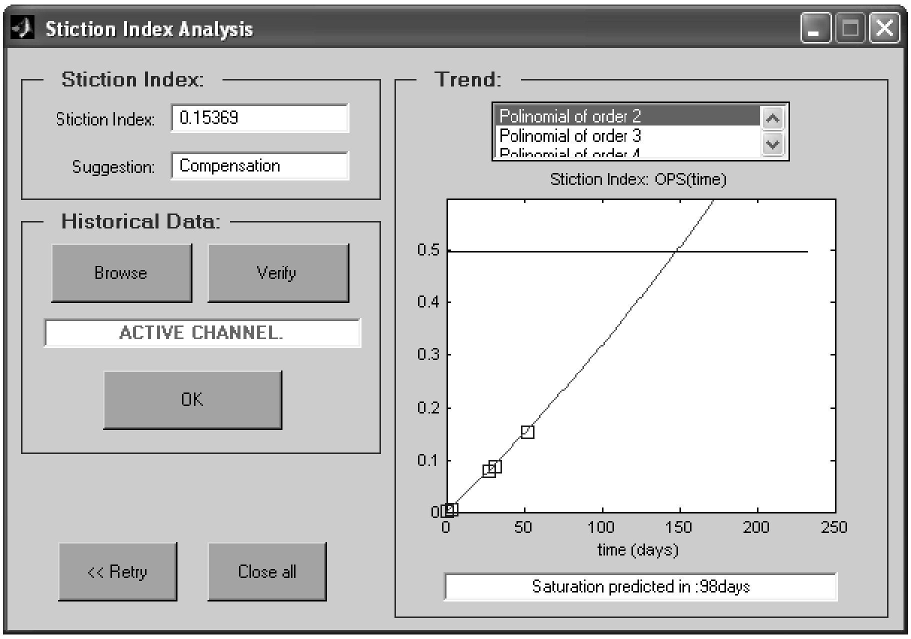
D4 = 0.38

	A	B	C	D	E
1	Year	Month	Day	FC140	
2	2006	1	19	0.200	
3	2006	1	22	0.220	
4	2006	2	15	0.380	
5	2006	2	19	0.404	
6					

Figure 5.20: Excel template for Stiction Index evaluation

The first three columns report year/month/day values, referred to the stiction index value indicated in column four: all numbers are automatically updated by the system. Users only need to open the Excel file relative to the loop under exam before starting this analysis.

The *Stiction Index* Window, which is opened by pressing the relative button in the *Report* Window (figure 5.14) is depicted in figure 5.21.



Stiction Index Analysis

Stiction Index:

Stiction Index: 0.15369

Suggestion: Compensation

Historical Data:

ACTIVE CHANNEL.

Trend:

Polinomial of order 2
Polinomial of order 3
Polinomial of order 4

Saturation predicted in :98days

Figure 5.21: *Stiction Index Window*

In a first section the stiction index value for collected data is evaluated and a suggestion between *Maintenance* and *Compensation* is reported. To update the excel file relative to the examined loop the user again needs to open it and to press the *Browse* button: only in case of a properly opened channel between Excel and MATLAB (*Verify* button) the *OK* button is enabled. By pressing it a graph, reporting historical and actual values of stiction index, is shown with a best-fit trend, chosen amongst functions listed in the *Trend* section. A message about the estimated period of time the valve needs to reach saturation is also shown in order to organize future maintenances.

Chapter 6

Extension to the MIMO processes

Over the last two decades, monitoring control loop performance has been addressed in several ways and several performance indices have been proposed (see [49] for a good survey). Different causes for low loop performance such as improper controller tuning, sensor faults, valve non-linearities have been identified [50, 51]. An important cause that demands attention in addition to these causes is the presence of interactions among loops. A key impact of the interaction on the loop performance is the propagation of the effects of other causes that deteriorate the loop performance, thereby corrupting other loops.

A multivariate ($MIMO$) system can be schematized with a matrix of transfer functions **P**, not necessary square, where diagonal elements represent the process transfer functions and off-diagonal elements (P_{ij}, $i \neq j$) represent the interaction transfer functions (see figure 6.1 for a reference scheme). When an excitation affects a loop *i*, some effect is also present on another loop *j* depending on the interaction transfer function P_{ij}.

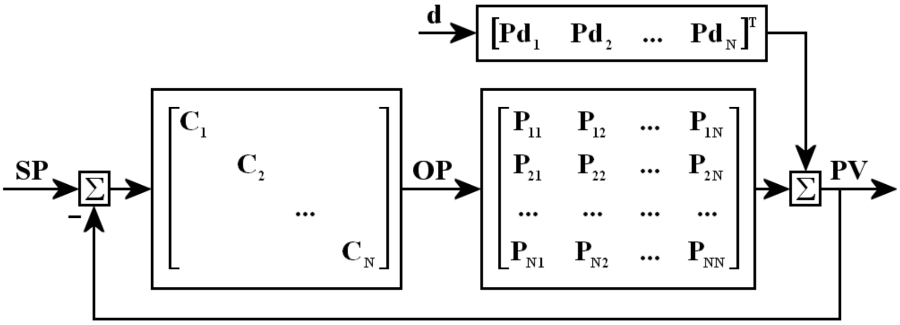


Figure 6.1: *Reference scheme for a* MIMO NxN *system*

[*] Portions of this chapter are published in [48]

The Relative Gain Array (RGA) is often used to describe the level of interaction among loops, for instance in [52]. However, it has two key limitations: (*i*) a model of the process must be known and consequentially the RGA measure depends on the model uncertainty [53] and (*ii*) RGA gives only a measure of stability once loops are closed and no indication on the real interaction among them.

Therefore, a novel approach is proposed, which does not use an explicit process model, but instead directly uses routine operating data. Once the loops that are suspected to interact are selected, the proposed method can be used to assess the level of interaction.

Among all types of excitations, the set-point excitation is most preferred since it allows us to obtain PV as a function only of **P** and matrix containing the controller transfer functions **C**. The effect of the disturbance transfer function **Pd**, which may cloud the interaction measure, is thus avoided.

6.1 Methodology to measure interactions

6.1.1 Interaction Index

The proposed technique is performed in the time domain and the error signal **e=SP-PV** is considered as input of the Hägglund technique ([1]) to built a modified IAE. It has to be reminded that the trend of IAE is composed by peaks which start at each zero crossing of the error signal and reach a relative maximum before the next zero crossing.

The use of this new variable allows us to magnify the difference between excitations (bigger peaks) and noise (smaller peaks) taking into account both the amplitude of the error and the duration between two consecutive zero crossings. Furthermore, the comparison of IAE peaks can be directly used to evaluate the amount of interaction between loops. To analyze only peaks due to excitations, a technique for the detection of outliers is applied [54]. This technique analyzes maxima of each peak; maxima of noise peaks generate a cluster from which the maxima of excitation peaks are excluded (figure 6.2).

In case of interactions, a set-point change will generate a peak in IAE trends in the two loops under examination almost at the same time. Considering the unknown time delays of the interaction path, it is impossible to establish the exact gap which occurs between the two peaks. To overcome this problem a time windows is chosen considering the duration of the source peak: defining t_0 as the time in which the set point change starts and t_1 to be the time at which the corresponding peak reaches its maximum, the time horizon t_h for the time window can be evaluated as follows:

$$t_h = t_0 + a \cdot (t_1 - t_0) \qquad \text{(Eq. 5-1)}$$

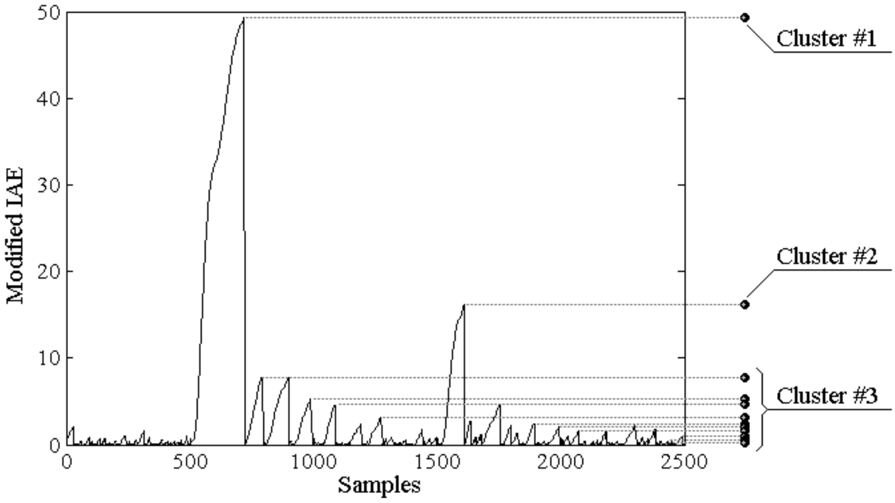


Figure 6.2: *Example of Excitation Peak and Cluster of Noise peaks*

where the parameter a is set equal to 4 by default. With this value, the time delay of the interaction (θ_I) is generally over-estimated:

- $t_1 - t_0$ is the time gap in which the controlled variable reaches the set-point value for the first time and corresponds to an overestimation of θ_P;
- under the hypothesis of similar values of θ_I and θ_P, the choice $a = 4$ allows one to obtain for a majority of cases a time horizon bigger than θ_I.

However the value of a can be changed easily by the operator to analyze the effect of the time window horizon on the interaction measure.

An interaction index (IM: Interaction Measure) can be evaluated analyzing the IAE trend over this user-specified time window:

$$IM_{i,j} = 1 - \frac{IAE_{SP}}{IAE_{SP} + IAE_I} \qquad \text{(Eq. 5-2)}$$

where SP and I indicate respectively the loop affected by the set point change and the interacting loop. IM lies between [0 1] and a strong interaction is associated with higher values.

A "fuzzy" interpretation of the interaction can be proposed for both the indexes as shown in table 6.1. Particularly important is the limit of 0.5; for the $IM_{i,j} > 0.5$ a set-point change in a loop *i* affect other loops more than loop *i* itself.

It should be noted that the this method has a limitation: it can not estimate correctly the interaction when set-point activity in a loop and a disturbance in another loop coincide. However, if different set-point changes are present in the data set, the application of the technique for each change reduces the importance of the problem.

Table 6.1: *Association between interaction measure and values of the two indexes*

IM	Interpretation
$[0 \quad 0.125]$	No Interaction
$[0.125 \quad 0.25]$	Low Interaction
$[0.25 \quad 0.375]$	Medium Interaction
$[0.375 \quad 0.5]$	High Interaction
$[0.5 \quad 1]$	Very High Interaction

6.1.2 Controller Performance Index

Applying the index described above the amount of interaction can be easily estimated; in case of high interaction a different index must be evaluated to understand if adopting a new tuning of implemented controllers will suffice to improve the performance or the adoption of a more advanced structure will be requested. For this purpose, a new Controller Performance Index (CPI) is here defined.

The CPI is proposed on the basis of the response to a set-point change. Given a set-point change, under *minimum variance control*, after $\theta_P + t_0$, the error immediately reaches zero, where t_0 is the time in which the set-point change starts and θ_P the time delay of the process. Suppose a minimum error e_{min} is associated with this case. If not under minimum variance control a residual error is still present until the controlled variable reaches the settling time. Denote the error in such a case by e_{tot}. The CPI is then defined as,

$$\text{CPI} = \frac{e_{tot} - e_{min}}{e_{tot} + e_{min}} \qquad \text{(Eq. 5-3)}$$

If e_{tot} is near to the minimum achievable, the controller has a good performance and CPI is near zero. If $e_{tot} \gg e_{min}$, the controller has a poor performance and CPI is near to one. Given the fact that the minimum variance controller is an idealistic case and of little practical use [55] and considering the presence of interaction, a threshold value of $CPI = 0.5$ is chosen. Below this value of CPI, retuning would be practically of little benefit. On the other hand, a low value of the CPI with a high value of IM implies that the present controller configuration yields good performance but unable to handle interactions. Therefore, a structural change may be necessary.

To evaluate the CPI, the time delay of the process θ_P must be estimated. The recorded response PV to the set-point change in closed loop can be approximated by a open loop response to a step-test $\tilde{PV}$. Choosing a second order model $\tilde{P}$ and varying its parameters, it is possible to find the best approximation in the least square sense. The obtained model will not have any physical meaning: it is used only to generate a good estimate of the time-delay (for the same reason the order of the model is not critical). Assuming a fixed value for the time delay $q = \theta_P/h$ with h sampling time and defining n the length of the data set, it is possible to generate the best approximation of PV in the least square sense:

$$PV(z^{-1}) = \frac{b_1 z^{-1-q} + b_2 z^{-2-q}}{a_1 z^{-1} + a_2 z^{-2} + 1} \cdot SP(z^{-}1) \qquad \text{(Eq. 5-4)}$$

$$PV_k = b_1 r_{k-1-q} + b_2 r_{k-2-q} - a_1 PV_{k-1} - a_2 PV_{k-2} \qquad \text{(Eq. 5-5)}$$

$$\underbrace{\begin{bmatrix} PV_{q+3} \\ PV_{q+4} \\ \vdots \\ PV_n \end{bmatrix}}_{\mathbf{y}} = \underbrace{\begin{bmatrix} -PV_{q+2} & -PV_{q+1} & SP_2 & SP_1 \\ -PV_{q+3} & -PV_{q+2} & SP_3 & SP_2 \\ \vdots & \vdots & \vdots & \vdots \\ -PV_{n-1} & -PV_{n-2} & SP_{n-1-q} & SP_{n-2-q} \end{bmatrix}}_{\mathbf{M}} \underbrace{\begin{bmatrix} a_1 \\ a_2 \\ b_1 \\ b_2 \end{bmatrix}}_{\mathbf{p}} \qquad \text{(Eq. 5-6)}$$

$$\mathbf{p} = (\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T \cdot \mathbf{y} \qquad \text{(Eq. 5-7)}$$

Varying the value of q, the array of the model parameters **p** can be estimated as shown in equation Eq. 5-7 and the approximating signal can be built using equation Eq. 5-5.

Among all models $\tilde{P} = f(q, \mathbf{p})$, the one that generates the lowest error in the least square sense is associated with the best estimation of θ_P.

6.2 Comparison with the RGA methodology

The advantage of using the IAE technique instead of the RGA can be easily demonstrated with an examination of a 2x2 $MIMO$ system in three different situations : a) a diagonally dominant system (equation Eq. 5-8); b) off-diagonal elements with bigger gains and slow dynamic (equation Eq. 5-9); c) off-diagonal elements with lower gains and fast dynamic (equation Eq. 5-10).

$$\begin{bmatrix} PV_1 \\ PV_2 \end{bmatrix} = \begin{bmatrix} \frac{1.5}{10s+1} & \frac{0.5}{60s+1} \\ \frac{-0.6}{40s+1} & \frac{2}{15s+1} \end{bmatrix} \cdot \begin{bmatrix} OP_1 \\ OP_2 \end{bmatrix} \qquad \text{(Eq. 5-8)}$$

$$\begin{bmatrix} PV_1 \\ PV_2 \end{bmatrix} = \begin{bmatrix} \frac{-0.6}{10s+1} & \frac{2}{60s+1} \\ \frac{1.5}{40s+1} & \frac{0.5}{15s+1} \end{bmatrix} \cdot \begin{bmatrix} OP_1 \\ OP_2 \end{bmatrix} \qquad \text{(Eq. 5-9)}$$

$$\begin{bmatrix} PV_1 \\ PV_2 \end{bmatrix} = \begin{bmatrix} \frac{1.5}{40s+1} & \frac{0.5}{15s+1} \\ \frac{-0.6}{10s+1} & \frac{2}{60s+1} \end{bmatrix} \cdot \begin{bmatrix} OP_1 \\ OP_2 \end{bmatrix} \qquad \text{(Eq. 5-10)}$$

In the first case (figure 6.3), both RGA ($RGA_{1,2} = 0.09$ and the IAE technique ($IM_{1,2} = 0.01$, $IM_{2,1} = 0.03$) indicate correctly a low interaction. Also in the second case (figure 6.4) both techniques give a correct interpretation of the problem indicating a strongly interacting system ($RGA_{12} = 0.91$, $IM_{12} = 0.52$ and $IM_{21} = 0.44$). On the contrary, in the third case (figure 6.5c) the RGA, which is not able to consider the effect of time constants, indicates wrongly the absence of interaction ($RGA_{12} = 0.09$); instead the IAE technique reveals correctly a medium-strong interaction ($IM_{12} = 0.28$, $IM_{21} = 0.39$).

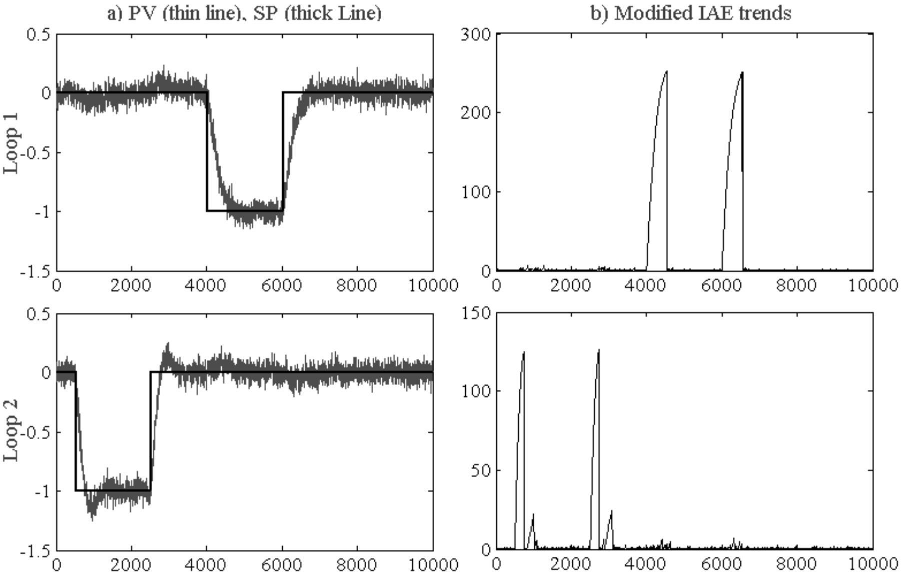


Figure 6.3: Diagonally dominant system: a) set-point (SP) and controlled variable (PV) values; b) IAE trends

6.3 Simulation Results

Several $MIMO$ systems were analyzed to demonstrate the effectiveness of the proposed techniques. Here three of them are described:

- The first example is a distillation column described in [56]; it has been chosen for its high level of interaction. Two different decentralized PI controllers will be tested.

- The second is a distillation column described in [57]; here a decentralized PI controller will be compared with the case of a steady state decoupler

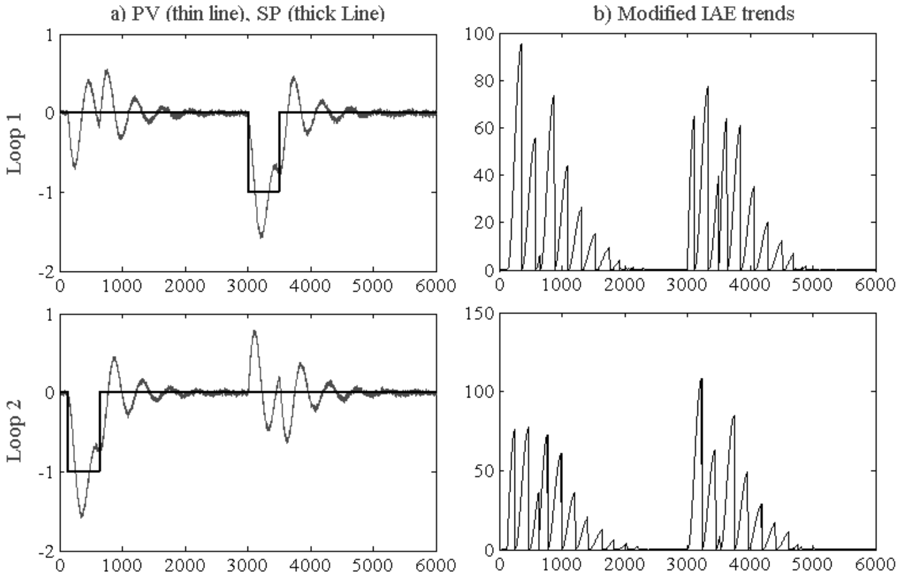


Figure 6.4: *Off-diagonal elements with bigger gains: a) set-point (SP) and controlled variable (PV) values; b) IAE trends*

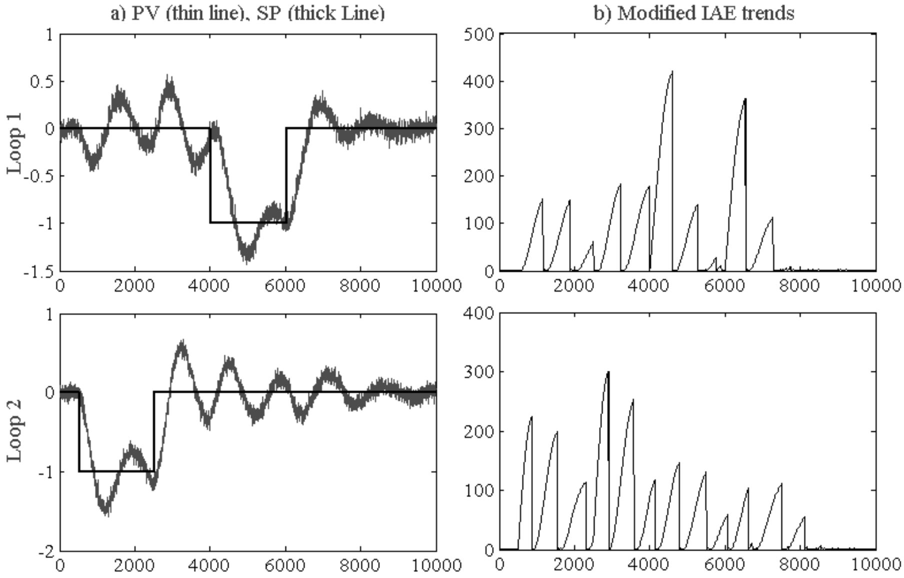


Figure 6.5: *Off-diagonal elements with faster dynamic: a) set-point (SP) and controlled variable (PV) values; b) IAE trends*

- The third is a distillation column described in [58]; in this case a decentralized PI controller will be compared with an MPC.

6.3.1 Marlin column

The scheme of the column is depicted in figure 6.6: the two potential manipulated variables are the reflux (F_R) and reboiler steam (F_V) flow rates; distillate and bottom compositions (x_D and x_B respectively) are the controlled variables; the feed composition (x_F), depending on upstream operations, cannot be adjusted *i.e.* it is a disturbance variable.

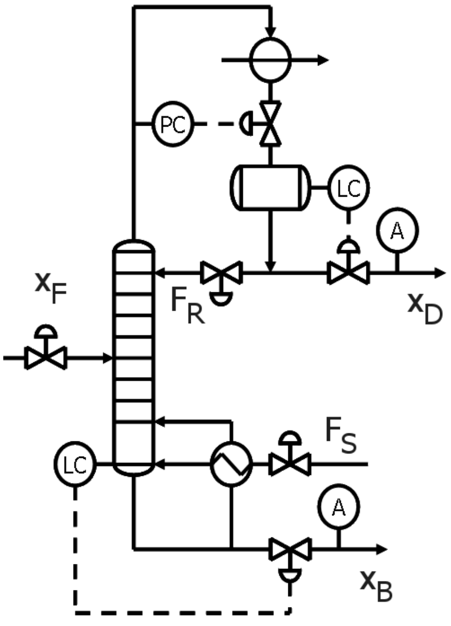


Figure 6.6: *Scheme of the Marlin 2x2 strongly interacting column*

The process transfer functions are:

$$\begin{bmatrix} x_D \\ x_B \end{bmatrix} = \begin{bmatrix} \dfrac{0.0747e^{-3s}}{12s+1} & \dfrac{-0.0667e^{-2s}}{15s+1} \\ \dfrac{0.1173e^{-3.3s}}{11.7s+1} & \dfrac{-0.1253e^{-2s}}{10.2s+1} \end{bmatrix} \cdot \begin{bmatrix} F_R \\ F_S \end{bmatrix} + \begin{bmatrix} \dfrac{0.07e^{-5s}}{14.4s+1} \\ \dfrac{1.3e^{-3s}}{12s+1} \end{bmatrix} \cdot x_F \qquad \text{(Eq. 5-11)}$$

It should be noted that the gains, time constant and time delay for diagonal and off-diagonal elements are similar indicating a strongly interacting system. Furthermore, the effect of x_F, being only a disturbance, will not be considered in this analysis, based only on set point changes. A detailed description of the process are reported in [56], the suggested PI decentralized controller are C_1 for loop 1 and C_2 for loop 2:

$$C_1 = \frac{10.4 \cdot (9s+1)}{9s+1}; \quad C_2 = \frac{-6.8 \cdot (6.1s+1)}{6.1s+1}; \qquad \text{(Eq. 5-12)}$$

Some set-point changes are introduced to analyze the presence of interaction as depicted in figure 6.7a; the correspondent IAE trends are reported in 6.7b.

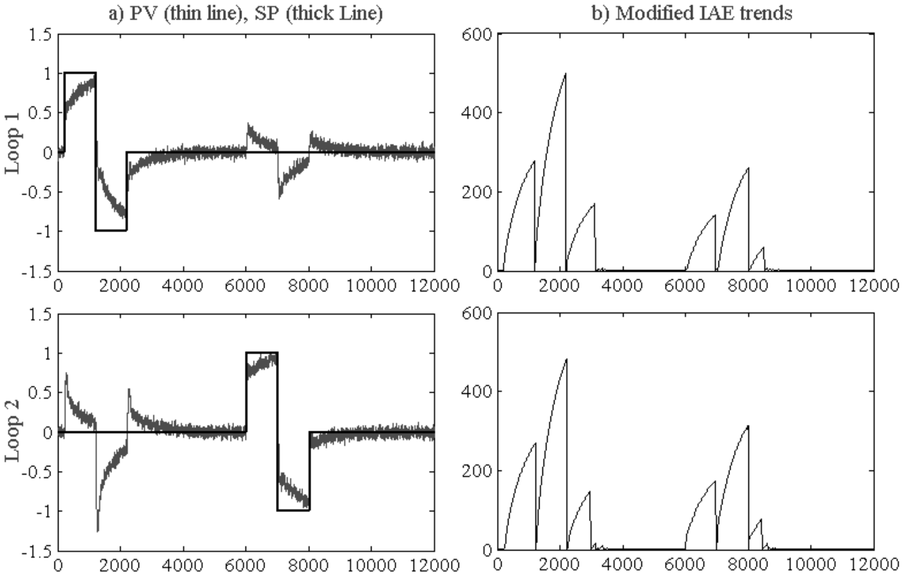


Figure 6.7: *Marlin Column: a) set-point (SP) and controlled variable (PV) values; b) IAE trends*

The interaction is obviously present: $IM_{1,2} = 0.47$ and $IM_{2,1} = 0.43$ indicate correctly that the system has an high interaction in both directions reproducing the expected result from the transfer functions analysis. The CPI index, equal to $[0.91 \div 0.92]$ suggests to adopt a different tuning to reduce the amount of interaction. Hence, the two PI controllers were changed as described in equation Eq. 5-13. The new effect of set-point changes is shown in figure 6.8.

$$C_1 = \frac{25 \cdot (5s+1)}{5s+1}; \quad C_2 = \frac{-10 \cdot (4s+1)}{4s+1}; \qquad \text{(Eq. 5-13)}$$

With the new controller tuning, the interaction between loops is substantially changed: loop 1 has again a big effect on loop 2 ($IM_{1,2} = 0.81$), however loop 2 do not affect anymore loop 1 ($IM_{2,1} = 0.09$), meaning that loop 2 is absorbing all excitations; this would be a good choice if the bottom composition was considered a slack variable. Lower values of CPI, equal to $[0.71 \div 0.75]$ indicate also a lower benefit

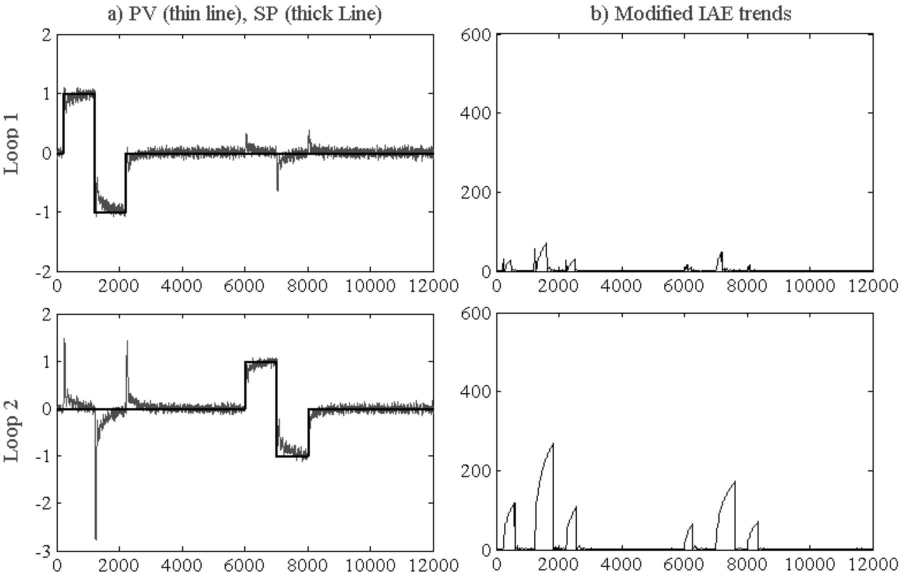


Figure 6.8: Marlin Column with new PI tuning set of parameters: a) set-point (SP) and controlled variable (PV) values; b) IAE trends

obtainable with a new tuning; a different structure for the controller (e.g. MPC) should be used to improve the performance.

6.3.2 Wood-Berry Column

The process transfer functions of the Wood-Berry column ([57]) are reported in equation Eq. 5-14; again a strong interaction is expected because of similar values of gains, time delay and time constant.

$$\begin{bmatrix} PV_1 \\ PV_2 \end{bmatrix} = \begin{bmatrix} \dfrac{12.8e^{-s}}{16.7s+1} & \dfrac{-18.9e^{-3s}}{21s+1} \\ \dfrac{6.6e^{-7s}}{10.9s+1} & \dfrac{-19.4e^{-3s}}{14.4s+1} \end{bmatrix} \cdot \begin{bmatrix} OP_1 \\ OP_2 \end{bmatrix} \qquad \text{(Eq. 5-14)}$$

Initially two decentralized PI controllers were adopted: $C_1 = 0.2 \cdot (10s + 1)/(10s)$ and $C_2 = -0.1 \cdot (15s+1)/(15s)$. The effect of set-point changes and IAE trends for this case are shown in figure 6.9 Applying the interaction technique, high values of indexes were obtained: $IM_{1,2} = 0.61$, meaning a very strong interaction from loop 1 to loop 2; $IM_{2,1} = 0.28$, meaning a weaker interaction from loop 2 to loop 1.

The CPI index, having in this case values under 0.7 suggests to adopt different controller structures rather than trying a new tuning; a steady state decoupler, whose

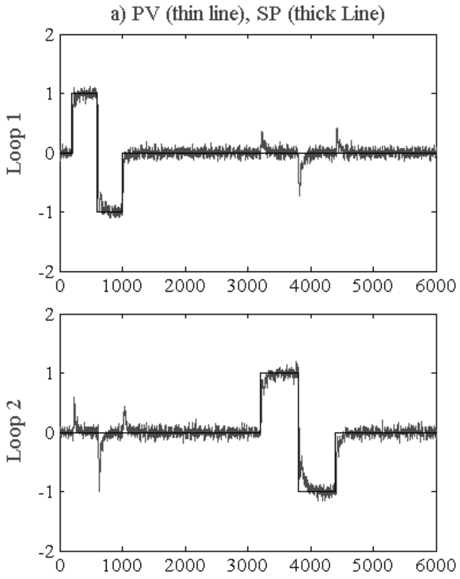


Figure 6.9: *Wood-Berry Column: a) set-point (SP) and controlled variable (PV) values; b) IAE trends*

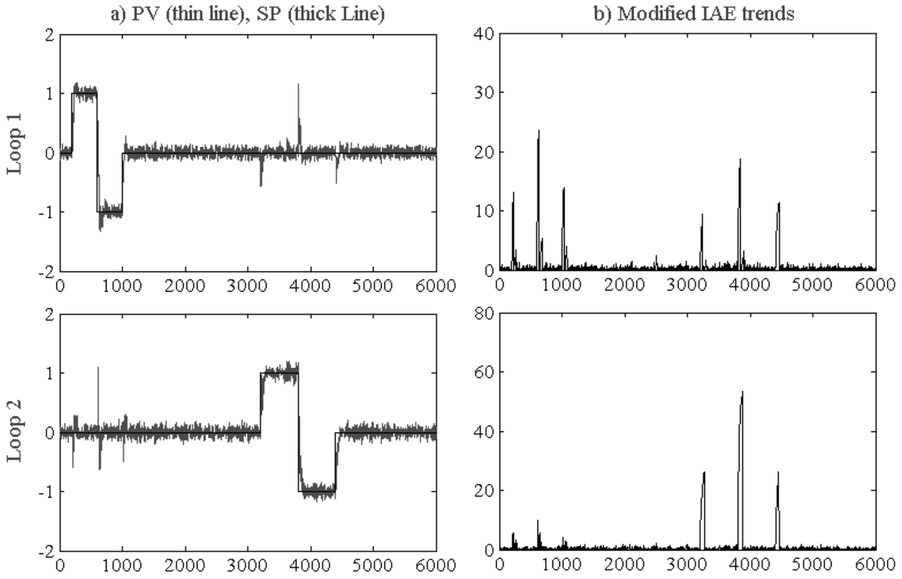


Figure 6.10: *Wood-Berry Column with a steady state decoupler: a) set-point (SP) and controlled variable (PV) values; b) IAE trends*

scheme is depicted in figure 6.11, was implemented.

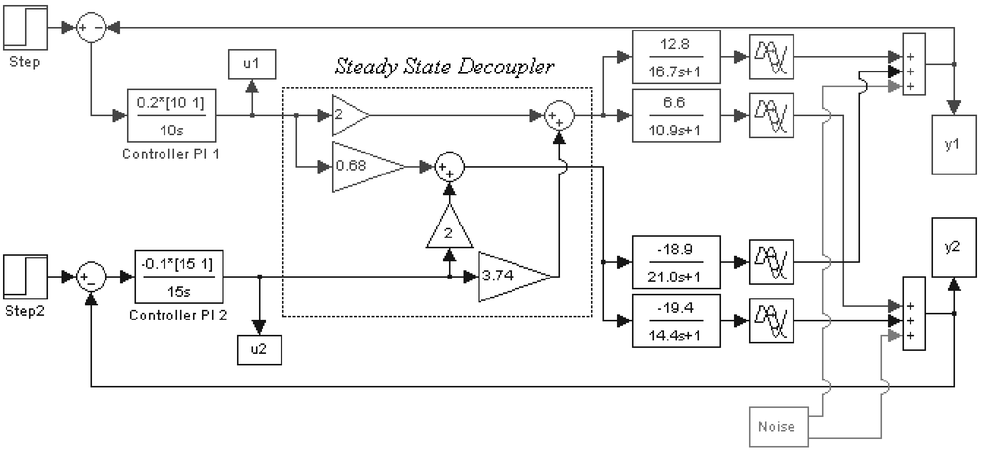


Figure 6.11: *$SIMULINK$ scheme of the Wood-Berry Column with decentralized PI controllers and a steady state decoupler*

With the decoupler a lower level of interaction is already visible in the IAE trends depicted in figure 6.10, where the peaks have lower amplitude than in the previous case. The confirmation can be found in the interaction indexes: $IM_{1,2} = 0.22$ and $IM_{2,1} = 0.20$ are related to a weak interaction in both directions, thus indicating a net improvement in performance with respect to the simple decentralized PI controllers.

6.3.3 Shell Column

The third system to be described is the *Shell benchmark problem* [58]; the process transfer functions are reported in equation Eq. 5-15:

$$\begin{bmatrix} PV_1 \\ PV_2 \\ PV_3 \end{bmatrix} = \begin{bmatrix} \frac{4.5e^{-27s}}{50s+1} & \frac{1.77e^{-28s}}{60s+1} & \frac{5.88e^{-27s}}{50s+1} \\ \frac{5.39e^{-18s}}{50s+1} & \frac{5.62e^{-14s}}{60s+1} & \frac{6.9e^{-15s}}{50s+1} \\ \frac{4.38e^{-20s}}{33s+1} & \frac{4.42e^{-22s}}{44s+1} & \frac{7.2}{19s+1} \end{bmatrix} \cdot \begin{bmatrix} OP_1 \\ OP_2 \\ OP_3 \end{bmatrix} \qquad \text{(Eq. 5-15)}$$

For this problem two solutions were analyzed: firstly a decentralized PI controller was implemented and secondly it was compared with the MPC proposed in [59] for which y_3 is considered as a slack variable. The response for the two cases to the same set-point changes are shown in figures 6.12 and 6.13 respectively.

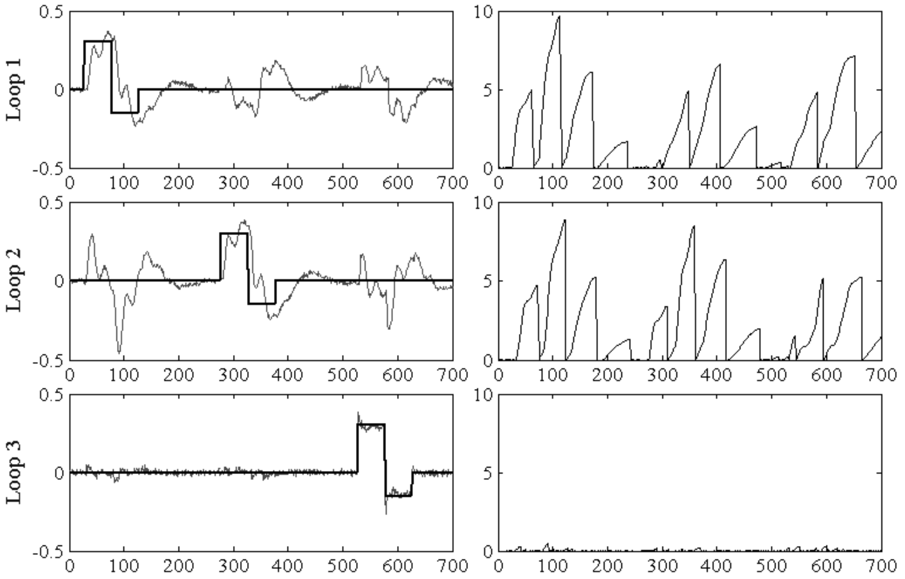


Figure 6.12: *Shell Problem with decentralized PI controllers: a) set-point (SP) and controlled variable (PV) values; b) IAE trends*

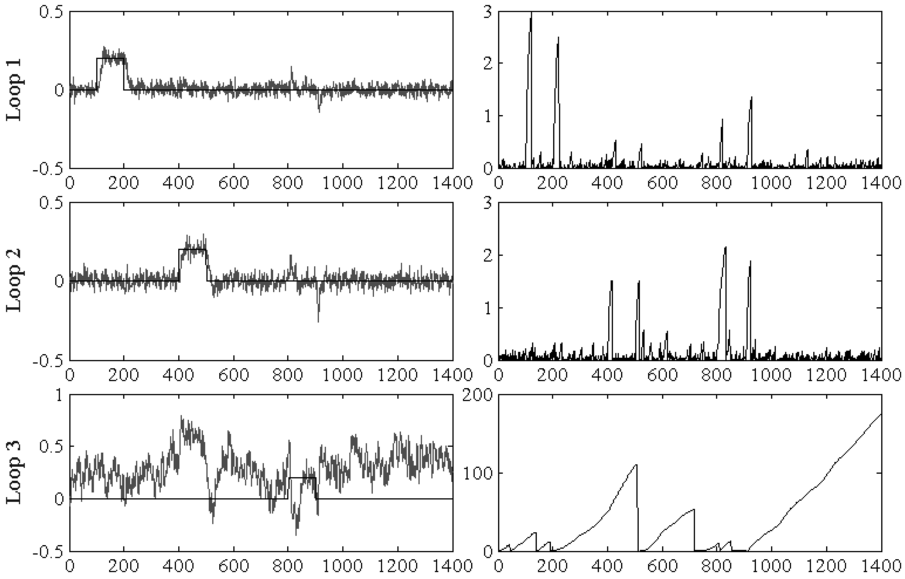


Figure 6.13: *Shell Problem with MPC: a) set-point (SP) and controlled variable (PV) values; b) IAE trends*

The two different situations are well represented by the IM measures:

$$\underbrace{IM=\begin{bmatrix}1 & .47 & .01\\ .46 & 1 & .01\\ .98 & .97 & 1\end{bmatrix}}_{dec.PI}\ ;\quad \underbrace{IM=\begin{bmatrix}1 & .05 & .6\\ .16 & 1 & .92\\ .0 & .0 & 1\end{bmatrix}}_{MPC} \qquad \text{(Eq. 5-16)}$$

It is clear that the solution with decentralized PI controllers is not performing satisfactorily; the first two loops are strongly interacting ($IM = 0.46 \div 0.47$); and loop 3 is affecting them ($IM = 0.97 \div 0.98$) without being affected ($IM < 0.1$). On the contrary with the MPC the first two loops are not interacting any more because the third loop is absorbing all excitations; the only residual interaction from loop 2 to loop 1 ($IM_{2,1} = 0.16$) has low importance.

6.4 Experimental setup

The IAE technique was experimentally applied on data obtained by a *4 tank process*, depicted with its scheme in figure 6.14.

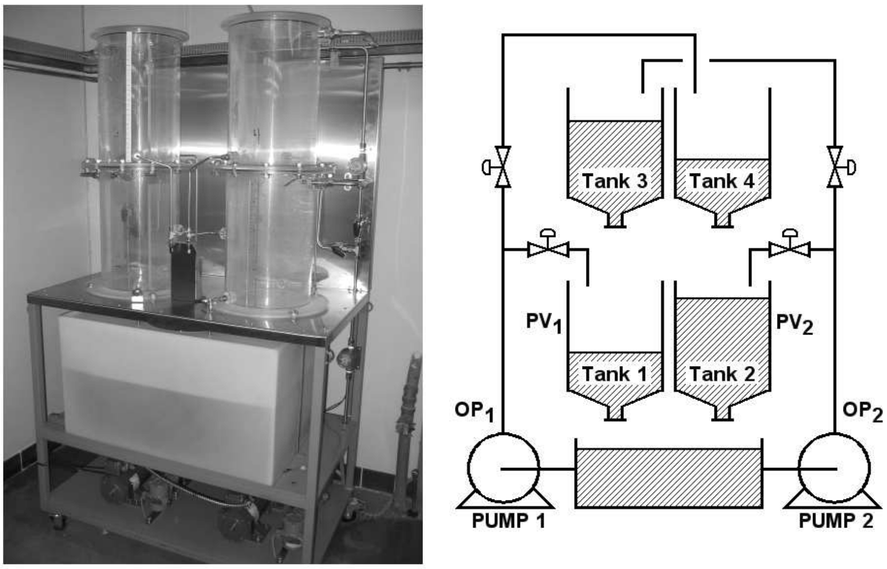


Figure 6.14: *Picture and scheme of the analyzed 4-tank system*

Two systems ([60]) can be considered and implemented: a minimum phase and a non-minimum phase system. The first configuration is obtained by adjusting the valves so that the flow from pump 1 is diverted in tank 1 and from pump 2 in tank 2; however maintaining the two three-way valve partially opened a small amount of flow is allowed from pump 1 to tank 4 and from pump 2 to tank 3, thus generating the interaction effect. The second configuration is obtained with pump 1 feeding tank 4 and pump 2 feeding tank 3; again with the two three-way valves partially opened in one direction an interaction effect can be generated by the flow from pump 1 to tank 1 and pump 2 to tank 2. For both cases the target is the control of the two levels in tank 1 and tank 2 by manipulating the inlet flowrates of the tanks adjusting the speed of the two pumps.

The first experiment consisted in adopting the minimum phase configuration and in changing the set-point of the two levels in order to evaluate the interaction effect. Set-point changes and controller variables are shown in figure 6.15 while IAE trends and interaction measure are shown in figure 6.16b for the minimum phase system.

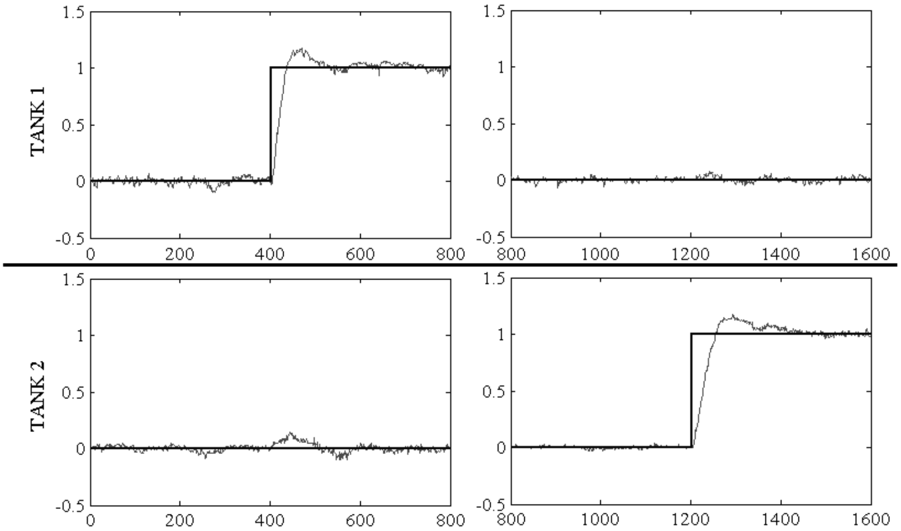


Figure 6.15: *Set point changes (thick lines) and controlled variables (thin lines) for the minimum phase case*

Analyzing the trend of the controlled variables, it is difficult to establish properly the level of interaction: an excitation with small amplitude is shown in tank 2 for a set-point change in tank 1, but it appears as a weak interaction. On the contrary, using the IAE trends the excitation in tank 2, because of its long duration, generates a significant peak similar to peaks showed in tank 1. Nevertheless, a set point change in tank 2 does not generate excitations in tank 1. This situation is well represented by the interaction matrix as shown in equation Eq. 5-17:

$$IM = \begin{bmatrix} 1 & 0.32 \\ 0.01 & 1 \end{bmatrix} \qquad \text{(Eq. 5-17)}$$

$IM_{1,2} = 0.32$ reveals the presence of an interaction of medium importance from tank 1 to tank 2; on the contrary $IM_{2,1} = 0.01$ implies that tank 2 is not affecting tank 1.

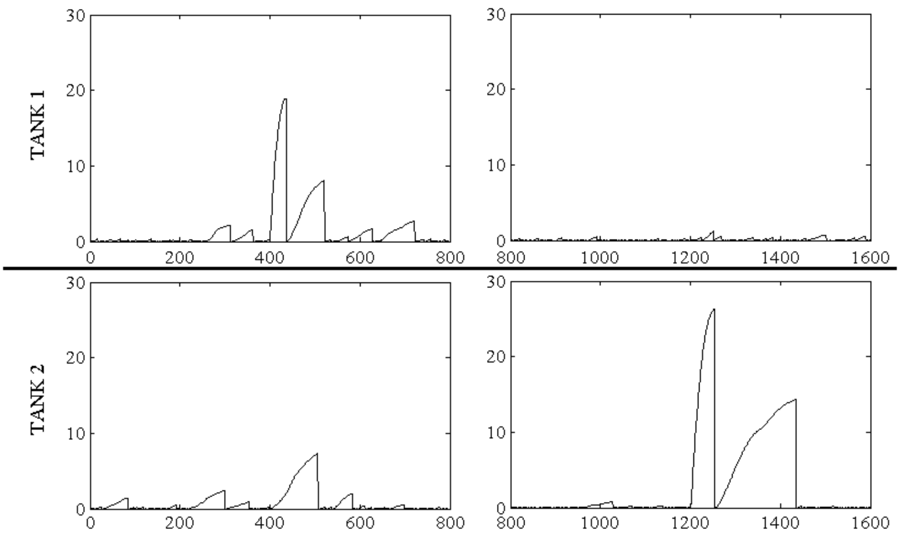


Figure 6.16: IAE *trends for the minimum phase case*

Set-point changes and controller variables are shown in figure 6.17 while IAE trends and interaction measure are shown in figure 6.18 for the non-minimum phase system.

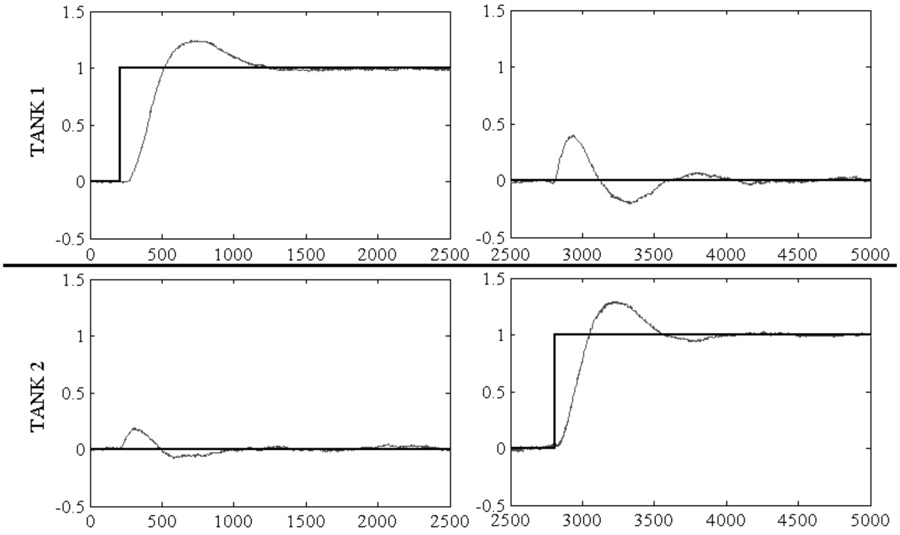


Figure 6.17: Set point changes (thick lines) and controlled variables (thin lines) for the non-minimum phase case

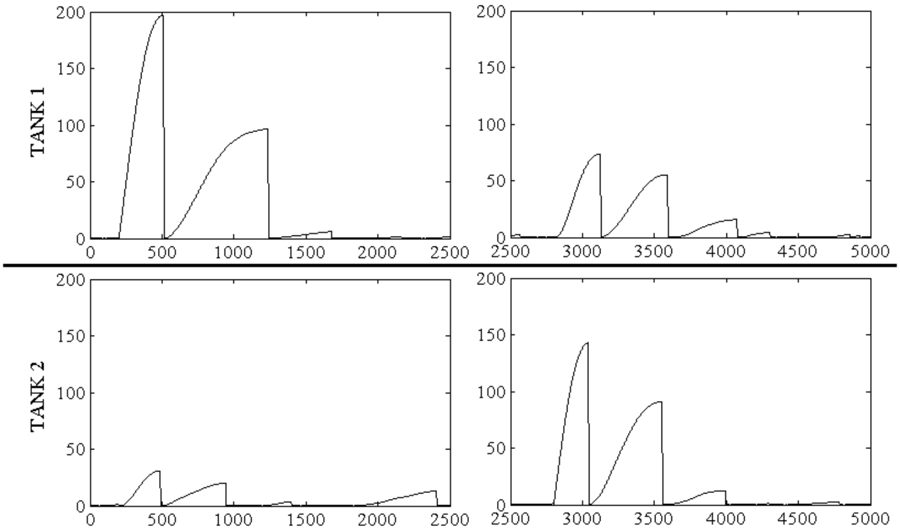


Figure 6.18: *IAE trends for the non-minimum phase case*

Again the analysis of IAE peaks reveals the presence of interaction which is stronger from tank 1 to tank 2 (equation Eq. 5-18).

$$IM = \begin{bmatrix} 1 & 0.11 \\ 0.36 & 1 \end{bmatrix} \qquad \text{(Eq. 5-18)}$$

To be noted that for the non-minimum phase case the highest interaction value is associated with $IM_{2,1}$ and the lowest with $IM_{1,2}$. This results is in agreement with the switch in the position of the feed for the two cases (pump 1 speed being controlled for level on tank 2 and pump 2 speed being controller for level on tank 1), indicating that the IAE technique gives a good interpretation of the presence of interaction. However, because of the time delay, the level of interaction between the two tanks appears to be bigger than in the previous case.

6.5 Development of a software tool

A software tool was designed to perform the interaction analysis, with the aim to be easily incorporated in the software package described in the previous chapter. This tool is based on four $MATLAB$ files:

- ***examiae.m***: the core of the program, where the evaluation of the interaction indexes is performed

- ***figurabase2.m***: the sequence of commands used to built the graphical interface with the user

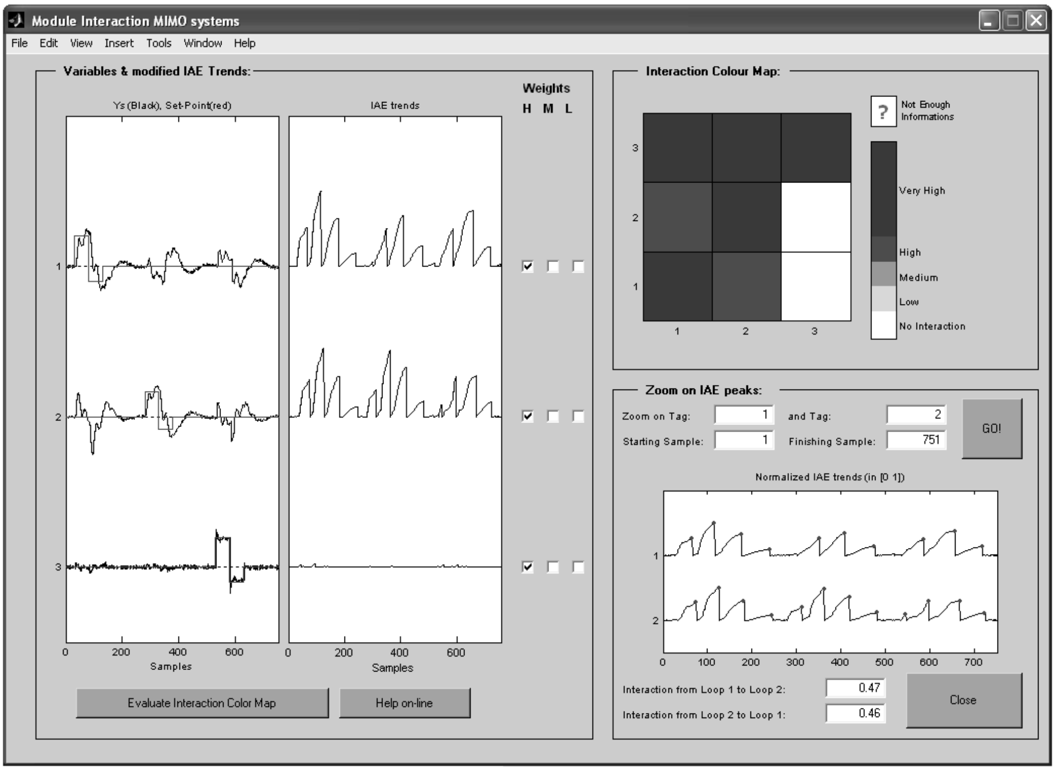
- ***makezoomonloops.m***: the routine to zoom on IAE trends displayed in the main window

- ***colormapagain.m***: the routine used to refresh the results of the proposed technique, showed as a color map: the darker the color the higher the interaction.

- ***checkb.m***: the routine used to associate a weight to the examined loops to discriminate them in order of importance

All these files must be stored in the same folder, however the user has only to call the main function, typing in the $MATLAB$ command window the following string:

[IM,IAE] = examiae(PVs,SPs)


Module Interaction MIMO systems
File Edit View Insert Tools Window Help
Variables & modified IAE Trends:
Ys (Black), Set-Point(red)
IAE trends
Weights
H M L
☑ □ □
☑ □ □
☑ □ □
1
2
3
0 200 400 600
Samples
0 200 400 600
Samples
Evaluate Interaction Color Map
Help on-line
Interaction Colour Map:
? Not Enough Informations
Very High
High
Medium
Low
No Interaction
1 2 3
Zoom on IAE peaks:
GO!
Normalized IAE trends (in [0 1])
0 100 200 300 400 500 600 700
Close

Zoom on Tag: 1 and Tag: 2

Starting Sample: 1 Finishing Sample: 751

Interaction from Loop 1 to Loop 2: 0.47

Interaction from Loop 2 to Loop 1: 0.46

Figure 6.19: *Graphical interface for the Interaction Analysis Tool*

The *examiae.m* files has to mandatory inputs: the controlled variables trends (PVs) and their relative set-points trends (SPs). Both of them must be organized in matrixes, whose element $X_{i,j}$ represent the i^{th} sample for loop j. The two outputs are IM, the interaction matrix, and IAE, the modified IAE trends; also the window depicted in figure 6.19 is displayed.

The main window is divided into three frames: *Variables and modified IAE trends*, *Interaction Color Map*, *Zoom and IAE peaks*. The first frame shows trends of set-points, controlled variables and modified IAE, with samples on abscissas and loop number on ordinates. The user can also use a weight for each loop on the basis of his experience by selecting the relative check box on the right side: high weights (H) corresponds to loop with high importance, low weights (L) to slack variables; also an intermediate situation (M, medium weight) is possible. When selecting a different weight for a loop, its relative IM indexes are scaled and the overall effect can be seen by pressing the button labelled *Evaluate Interaction Color Map*.

The second frame shows the interaction matrix using a color map where darker colors are symptom of higher interaction: where no set-point changes are detected the system displays a question mark, being unable to evaluate properly the interaction index for this case.

The last frame shows the zoomed IAE trend of the loops: the user has only to select the two loops to be depicted by introducing their number in the white fields and to define the range of samples to be shown; by pressing the GO! button the zoom is performed. This frame was inserted to help the user in examining more in detail the peaks in IAE trends highlighted by a red dot, namely the ones used by the technique to calculate the interaction index. Under the graph two fields show the IDT measure for the chosen loops.

Appendix A

Stiction techniques

A.0.1 Cross correlation algorithm

The two cases the Cross-Correlation function C_{XY} between OP and PV can be easily evaluate using:

$$C_{XY}(d) = \frac{\sum_{i=1}^{N}[(PV(i) - \overline{PV})] \cdot [(OP(i-d) - \overline{OP})]}{\sqrt{\sum_{i=1}^{N}[(PV(i) - \overline{PV})]} \cdot \sqrt{\sum_{i=1}^{N}[(OP(i-d) - \overline{OP})]}} \qquad \text{(Eq. } A\text{-1)}$$

Horch assumes that in case of an external periodical perturbation the two variables show up as pure sinusoidal waves and in case of stiction the trend in OP and PV are respectively a triangular and a square wave. These signals generate an odd cross correlation C_{XY} in case of stiction ($\phi = \pi/2$) and an even C_{XY} in case of oscillating disturbances ($\phi = \pi$). This assumption allows an automatic detection of the two cases, through the computation of the two parameters:

$$\Delta\tau = \frac{|\tau_I - \tau S|}{\tau_I + \tau S}; \qquad \Delta\rho = \frac{|r_0 - r_{Max}|}{r_0 + r_{Max}} \qquad \text{(Eq. } A\text{-2)}$$

The two parameters can be calculate from the C_{XY}: τ_I e τ_S represent the two abscissas, closest to zero, where $C_{XY} = 0$; r_0 is the value of C_{XY} in zero and r_{Max} is the maximum value of C_{XY}.

A.0.2 Bicoherence algorithm

The Bispectrum is based on the discrete fourier transform of the signal, indicated in the sequel with Y. By defining a couple of frequencies (f_1,f_2), the Bispectrum $B(f_1, f_2)$ and the Bicoherence $bic^2(f_1, f_2)$ functions, can be calculated:

$$B(f_1, f_2) = E[Y(f_1)Y(f_2)Y^*(f_1 + f_2] \qquad \text{(Eq. } A\text{-3)}$$

$$bic^2(f_1, f_2) = \frac{|B(f_1, f_2)|^2}{E[|Y(f_1)Y(f_2)|^2]E[|Y(f_1 + f_2|^2]} \qquad \text{(Eq. } A\text{-4)}$$

where $E|Y|$ is expectation value of Y and Y^* is conjugate complex of Y. Each point of $bic^2(f_1, f_2)$, plotted against f_1 and f_2 in a three dimensional plot, measures the interaction between frequencies f_1 and f_2. More details can be found in [29].

With the bic^2 it is possible to make a test on gaussianity and linearity of the signal using respectively the *Non Gaussian Index NGI* and the *Non Linear Index NLI*:

$$NGI \triangleq \overline{bic^2} - \overline{bic^2_{Cr}} \qquad \text{(Eq. } A\text{-5)}$$

$$NLI \triangleq |bic^2_{max} - (\overline{bic^2} + 2\sigma_{bic^2})| \qquad \text{(Eq. } A\text{-6)}$$

where $\overline{bic^2}$ and σ_{bic^2} are the mean value and the standard deviation of the bicoherence; furthermore the factor $\overline{bic^2_{Cr}}$ depends on the probability of obtaining false detection assuming bicoherence χ^2 distributed.

A.0.3 Relay algorithm

The sinusoidal approximation C_S is:

$$C_S = A_S \cdot sin(\omega_S t + \phi_S) \qquad \text{(Eq. } A\text{-7)}$$

Parameters A_S, ω_S and ϕ_S are obtained by implementing the Matlab function “fminsearch.m”, which uses the *Simplex* method. The starting point is calculated on the basis of recorded data: amplitude and frequency are the same of the recorded oscillation; the initial phase is chosen equal to zero.

The triangular approximation C_T is also very simple:

$$C_T = \begin{cases} A_T(m_1 t + q_1) & n \leqslant \overline{n} \\ A_T(m_2 t + (m_1 - m_2)t(\overline{n}) + q_1) & n > \overline{n} \end{cases} \qquad \text{(Eq. } A\text{-8)}$$

where $\overline{n}$ is the vertex of the triangle. For points $n_i \leqslant \overline{n}$ the approximation curve is a straight line with coefficient $m_1 > 0$ and for $n_i > \overline{n}$ the approximation curve is a straight line with coefficient $m_2 < 0$. The intercept of the second straight line is chosen to have the same value for the two straight line on the vertex of the triangle. The term A_T is only a scale factor; all the parameters A_T, m_1, m_2, q_1 are easily obtained with a linear least square technique (LLS).

More complex is the relay wave approximation: it starts with the division of the recorded oscillation into three intervals defined by two numbers n_1 and n_2. The corresponding three part of the recorded curve will be called in the sequel $y^{(1)}_{re}$, $y^{(2)}_{re}$, $y^{(3)}_{re}$. The division in three intervals allows to build a good algorithm for the generation of the best approximating relay wave-shape:

1. the procedure chooses a value for n_1 and n_2 subject to $n_2 > n_1$

2. the optimization modulus (based again on the Simplex method) for each iteration fixes a value for the two parameters that characterize the relay wave: the time constant τ and the delay θ.

3. the use of these parameters allows to obtain an approximating curve with unitary amplitude in the first interval :

$$\hat{y}_1 = 1 - e^{-\frac{t-\theta}{\tau}} \qquad \text{(Eq. } A\text{-9)}$$

4. The amplitude A_1 that allows to obtain the best approximating curve in the first interval (y_1) is calculated with the LLS technique:

$$A_1 = (\hat{y}_1^T \hat{y}_1)^{-1} \hat{y}_1 y_{re}^{(1)} \rightarrow y_1 = A_1 \cdot \hat{y}_1 \qquad \text{(Eq. } A\text{-10)}$$

5. The approximating curve in the second interval has the same τ of y_1; the delay in this case (θ_2) is evaluated in order to obtain $\hat{y}_2(n_1) \equiv \hat{y}_1(n_1)$; with this imposition the second part of the approximating curve with unitary amplitude is:

$$\hat{y}_2 = -1 + e^{-\frac{t-\theta_2}{\tau}} \qquad \text{(Eq. } A\text{-11)}$$

6. The calculation of the best-fitting amplitude A_2 in this case must be subjected to $y_2(n_1) \equiv y_1(n_1)$; this problem is resolved by equation Eq. A-12 and Eq. A-13. The value of A_2 is then evaluated again with a LLS method (equation Eq. A-14):

$$\tilde{y}_2 = \hat{y}_2 - \hat{y}_2(n_1) \qquad \text{(Eq. } A\text{-12)}$$

$$\tilde{y}_{re}^{(2)} = y_{re}^{(2)} - y_{re}^{(2)}(n_1) \qquad \text{(Eq. } A\text{-13)}$$

$$A_2 = (\tilde{y}_2^T \tilde{y}_2)^{-1} \tilde{y}_2 \tilde{y}_{re}^{(2)} \rightarrow y_2 = A_2 \cdot \hat{y}_2 \qquad \text{(Eq. } A\text{-14)}$$

7. The third part of the approximating function is evaluated in the same manner: it is fixed a value θ_3 as at step 5 and an amplitude A_3 as at step 6. In this way we can obtain also y_3.

8. The global approximating curve is the union of the three approximating parts: $y_{ap} = [y_1^T \; y_2^T \; y_3^T]^T$. A global scale factor A_{Tot} is evaluated with LLS between y_{ap} and the recorded curve in order to make the best possible approximation.

9. If y_{ap}, now evaluated with the new amplitude, generates a lower error than y_{ap} at the previous iteration, the system uses the new curve as the best-fit approximation C_R; else the C_R at previous iteration is maintained.

10. The Simplex modulus makes a first check: if it has not reached the convergence, it changes values for τ and θ and the procedure restarts from step 3.

11. If Simplex method is converged or has reached the maximum number of allowed iterations, the procedure makes the second check: if n_1 and n_2 have not assumed all the possible values the procedure restarts from step 1, else the best approximating curve is chosen as C_R.

Appendix B

Simplex Method

The $Simplex - \{N\}$ is a region of a N dimensions space with $N+1$ vertexes, $N \cdot (N-1)/2$ sides and ${}_{k+1}C_{i+1}$ surfaces, where the binomial coefficient ${}_{k+1}C_{i+1}$ can be expressed as:

$$ {}_{k+1}C_{i+1} = \binom{k+1}{i+1} \triangleq \frac{(k+1)!}{[(k+1)-(i+1)]! \cdot (i+1)!} \qquad \text{(Eq. } B\text{-1)} $$

To built the $simplex - \{N\}$, $N+1$ points are considered (P_0,P_1,...,P_N); associated values of the function to be minimized will be indicated in the sequel as z_0,z_1,...,z_N, where $z_i = f(P_i)$. Furthermore $\{\cdot\}_L$ and $\{\cdot\}_H$ suffixes will be used respectively for minimum and maximum values of z array; C_p will represent the barycenter calculated with all z values, being only excluded z_H value; $[P_i; P_j]$ will indicate the Euclidean distance between P_i and P_j.

For each step of the simplex algorithm P_H, the point relative to the maximum value in z array, is substituted with a new point, calculated by the application of three basic transformations: *reflection*, *expansion* and *contraction*. Reflected point P^* is represented by following coordinates:

$$ P^* = (1+\alpha) \cdot C_P - \alpha \cdot P_H \qquad \text{(Eq. } B\text{-2)} $$

where α is a positive constant named reflection coefficient. Applying this transformation P^* point belongs to the line which connects P_H and C_P at a distance from extremes function of coefficient α:

$$ [P^*; C_P] = \alpha \cdot [P_H; C_P] \qquad \text{(Eq. } B\text{-3)} $$

In case of $z^* \in [z_L \div z_H]$, P_H is substituted by its reflection P^*, a new $simplex - \{N\}$ is built and the procedure starts the next iteration.

On the contrary, if $z^* < z_L$, reflection generates a new minimum value; in this case next step is the expansion of P^* point on P^{**}, whose coordinates are:

$$P^{**} = (1-\gamma) \cdot C_P + \gamma \cdot P^* \qquad \text{(Eq. } B\text{-4)}$$

where the expansion coefficient γ must be chosen strictly greater than unity; similar considerations to Eq. B-3 can be done. In case of $z^{**} < z_L$, again a new minimum is found; hence P_H will be substituted by P^{**} and the procedure is restarted. Instead, in case of $z^{**} \geqslant z_L$, expansion has not reached its purpose of finding a new minimum: P_H will be substituted by P^* before starting with a new expansion with a different γ value.

Eventually, the following case must be considered: in reflecting P_H the new point P^* obeys to Eq. B-5:

$$P^* \ : \ z^* > z_i \qquad for\ i = 0..N \qquad s.t.\ i \neq H \qquad \text{(Eq. } B\text{-5)}$$

in this case a contraction must be performed, substituting P_H with P^{***} evaluated as follows:

$$P^{***} = \beta \cdot P_{Max} + (1-\beta) \cdot C_P \qquad \text{(Eq. } B\text{-6)}$$

where P_{Max} is the point whose value is $max\{z^*, z_H\}$; contraction coefficient β lies in the range $[0 \div 1]$. In case of $z^{***} \geqslant min\{z_H; z*\}$, P_H point will be substituted with P^{***} and the procedure restarts, otherwise all points will be substituted with $(P_i + P_L)/2$ before passing to the next iteration of the algorithm.

When the $simplex - \{N\}$ collapses on a point, considering a tolerance correction value to perform finite steps, a local minimum is reached (C_P of last built $simplex - \{N\}$) and the procedure is stopped.

This algorithm can be easily adapted to described procedures (chapter 5) by using the *fimnsearch.m* Matlab function (Optimization Toolbox), whose syntax is:

$$[x, fval, exitflag, out] = fminsearch(@myfun, x_0, opt, var_M) \qquad \text{(Eq. } B\text{-7)}$$

This function does an unconstrained nonlinear optimization using an initial guess (starting point x_0, namely an array with N estimated values for a N-variables function); minimum value of the function and correspondent variables values are stored respectively in $fval$ and x, being the former a scalar and the latter an array with same dimensions of x_0.

The function to be analyzed is recalled by Matlab function handle @ followed by a subroutine name (as for instance *myfun*), which must be built adopting the syntax:

$$function\ f_K = myfun(x_K, var_K); \qquad \text{(Eq. } B\text{-8)}$$

The body of *myfun* subroutine contains all commands to calculate the value of the function f_K using variables stored in the array x_K, automatically overwritten by the system with x_0 (being the initial guess x_0 updated at each iteration). Other variables to be used to evaluate the function can be passed from main program (var_M) to the *myfun* subroutine (var_K). Among outputs of *fminsearch.m*, *exitflag* variable gives information about the exit condition of the algorithm: a value equal to 1 is assumed in case of convergence; equal to 0 in case of maximum number of iteration exceeded; equal to -1 if interrupted internally by commands in *myfun*. *out* is a class variable containing: the name of the algorithm used, the number of performed function and iterations evaluations, an exit message about convergence of the algorithm.

Matlab *fminsearch* provides also interesting options the user can adopt (variable *opt* in Eq. B-7):

- **Display**: in case of ‘OFF’ value no output is displayed in Matlab command window; ‘iter’ will display an output message at each iteration, ‘final’ the final output and ‘notify’ gives back a message only when maximum number of function evaluations or iteration is exceeded.

- **FunValcheck**: because the *fminsearch.m* function can only be applied on real numbers, this option allow to give a warning in case of complex numbers (choosing ‘on’) or not (choosing ‘off’).

- **MaxFunEvals** and **MaxIter**: respectively the maximum number of function and maximum number of iteration allowed; those two options can be used to decrease calculation time were accuracy of the result has low importance (large tolerance on x values).

- **Tolfun** and **Tolx**: respectively tolerances on minimum value of the function and values of variables; in proposed procedure (chapter 5), the $Tolx$ option has been adopted, thus choosing to stop the simplex algorithm when process and disturbance parameters difference in two consecutive iterations is less than the defined $Tolx$.

An example of the syntax for *opt* variable is reported in Eq. B-9, using the routine *optimset*; particularly following options are chosen: no messages in the command window, a maximum of 200 iteration and a tolerance of 0.01 on parameters value.

$$opt = \text{optimset('Display','off','MaxIter',200,'Tolx',0.01);} \qquad \text{(Eq. } B\text{-9)}$$

To use constraints on process parameters (time delays and time constants strictly positive), an *if-then* sequence can be used inside *myfun*subroutine, forcing the system to produce $f_K = \infty$ for not allowed values of parameters; the algorithm will exclude this region of the N dimensional space during $simplex - \{N\}$ reflection, expansion or contraction.

Bibliography

[1] T. Hägglund, A control loop performance monitor, Control Engineering Practice 3 (1995) 1543–1551.

[2] N. F. Thornhill, D. E. Seborg, New directions in control system performance monitoring, T-Mo-A11, Session of 15th IFAC World Congress, Barcelona (Spain).

[3] B. Huang, S. L. Shah, Control loop performance assessment: Theory and applications, Springer-Verlag London.

[4] K. J. Aström, T. Hägglund, Pid controllers: Theory design and tuning, ISA, Research Triangle Park, N.C.(USA).

[5] G. Marchetti, C. Scali, Use of modified relay techniques for the design of model based controllers for chemical processes, Industrial & Engineering Chemistry - Research 39(9) (2000) 3325–3334.

[6] D. W. Clarke, Sensor, actruator, and loop validation, IEEE Control System Magazine 15(4) (1995) 39–45.

[7] M. Rossi, C. Scali, M. Amadei, Development of a technique for performance evaluation of industrial controllers, IFAC-Adchem'03 Int. Conf., Hong Kong (HK), Preprints Volume I, session 1.3 (2003) 183–188.

[8] M. Rossi, C. Scali, Una tecnica automatica per individuare la presenza di attrito negli attuatori, Automazione e Strumentazione 52(2) (2004) 91–97.

[9] M. Rossi, C. Scali, Automatic detection of stiction in actuators: A technique to reduce the number of uncertain cases, Proceedings (on CD) of DYCOPS'07 - 7th IFAC Int. Symp.: "Dynamics and Control of Process Systems", Cambridge (USA) Paper #157.

[10] N. F. Thorhill, T. Hägglund, Detection of multiple oscillations in control loops, Control Engineering Practice 5(10) (1997) 1343–1354.

[11] T. J. Harris, Assessment of control loop performance, Canadian Journal of Chemical Engineering 70 (1967) 1186–1197.

[12] N. F. Thorhill, B. Huang, H. Zhang, Detection of multiple oscillations in control loops, Journal of Process Control 13 (2003) 91–100.

[13] J. G. Ziegler, N. B. Nichols, Optimum settings for automatic controllers, ASME 65 (1942) 433–444.

[14] G. M. Jenkins, G. C. Reinsel, Time Series Analysis: Forecasting and Control, third edition,Prentice Hall, 1994.

[15] A. K. Tangirala, S. L. Shah, N. F. Thornhill, Pscmap: a new tool for plant-wide oscillation detection, Journal of Process Control 15(8) (2005) 931–941.

[16] M. Rossi, C. Scali, A comparison of techniques for automatic detection of stiction: Simulation and application to industrial data, Journal of Process Control 15(5) (2005) 505–514.

[17] T. Hägglund, A friction compensator for pneumatic control valves, Journal of Process Control 12 (2002) 897–904.

[18] A. Kayihan, F. J. DoyleIII, Friction compensation for a process control valves, Control Engineering Practice 8 (2000) 799–812.

[19] B. Armstrong-Hélouvry, P. Dupont, C. C. de Wit, A survey of models, analysis tools and compensation methods for the control of machine with friction, Automatica 30 (1994) 1083–1138.

[20] D. Karnopp, Computer stimulation of stic-slip friction in mechanical dynamical systems, ASME, Journal of Dynamic Systems, Measurement and Control 10 (1985) 100–103.

[21] M. A. A. S. Choudhury, S. L. Shah, N. F. Thornhill, A data-driven model for valve stiction, IFAC-Adchem'03 Int. Conf., Hong Kong (HK); Preprints Volume I, session 2.2 (2003) 261–266.

[22] Q. Taha, G. A. Dumont, M. S. Davies, Detection and diagnosis of oscillation in control loops, Proc. of 35th Conference on Decision and Control, Kobe, (Japan) (1996) 2432–2437.

[23] K. Manabu, H. Maruta, H. Kugemoto, K. Shimizu, Practical model and detection algorithm for valve stiction, IFAC Dycops-7th International Conference, Cambridge (USA); paper #54.

[24] A. Horch, A. J. Isaksson, A method for detection of stiction in control valves, IFAC Wokshop on On-Line Fault Detection and Supervision in the Chemical Process Industry, Lyon (France) (1998) 257–262.

[25] A. Horch, A simple method for detection of stiction in control valves, Control Enginnering Practice 7 (1999) 1221–1231.

[26] A. Horch, A. J. Isaksson, Detection of valve stiction in integrating processes, European Control Conference, Porto (Portugal) (2001) 1327–1332.

[27] R. Rengaswamy, T. Hägglund, V. Venkatasubramanian, A qualitative shape analysis formalism for monitoring control loop performance, Eng. Appl. Artif. Intell. 14(1) (2001) 23–33.

[28] A. Stenman, F. Gustafsson, K. Forsman, A segmentation-based method for detection of stiction in control valves, Int. J. Adapt. Control Signal Process 17 (2003) 625–634.

[29] M. A. A. S. Choudhury, S. L. Shah, N. Thorhill, Diagnosis of poor control loop performance using higher order statistics, Automatica 40(10) (2004) 1719–1728.

[30] G. Mori, Monitoraggio di sistemi di controllo: Individuazioni di anomalie dai dati d'impianto, Master Thesis in Chemical Engineering, University of Pisa (in Italian).

[31] W. B. Collis, P. R. White, J. K. Hammond, Higher-order spectra: the bispectrum and the trispectrum, Mechanical System and Signal Processing 12(3) (1998) 375–394.

[32] W. L. Luyben, Getting more information from relay feedback tests, Industrial Engineering and Chemistry Research 40(20) (2001) 4391–4402.

[33] T. Thyagarajan, C. C. Yu, Improved autotuning using the shape factor from relay feedback, Ind. Eng. Chem. Res. 40(10) (2003) 4425–4440.

[34] Y. Yamashita, An automatic method for detection of valve stiction in process control loops, Control Engineering Practice In press.

[35] C. Scali, M. Rossi, An automatic system for closed loop performance monitoring, extended version, AIDIC Conference Series , ISBN 0390-2358 7 (2005) 321–330.

[36] M. Rossi, C. Scali, Identification of process and disturbance dynamics to improve controller performance, Journal of Process Control, Submitted.

[37] D. B. Ender, Process control performance: Not as good as you think, Control Engineering 40(10) (1993) 180–190.

[38] C. C. Yu, Autotuning of PID controllers, Springer Verlag, 1999.

[39] J. A. Nelder, R. Mead, A simplex method for function minimization, Computer Journal 7 (1965) 308–313.

[40] J. C. Lagarias, J. A. Reeds, M. H. Wright, P. E. Wright, Convergence properties of the nelder-mead simplex method in low dimensions, SIAM Journal of Optimization 8(1) (1998) 112–147.

[41] S. Skogestad, Simple analytic rules for model reduction and pdi controller tuning, Journal of Process Control 13 (2003) 291–309.

[42] T. Hägglund, Automatic detection of sluggish control loop, Control Engineering Practice 7 (1999) 1505–1511.

[43] T. J. Harris, Assessment of control loop performance, The canadian Journal of Chemical Engineering 67 (1989) 856–861.

[44] W. K. Ho, C. C. Hang, J. H. Zhou, Performance and gain and phase margins of well-known pi tuning formulas, IEEE Trans. on Control System Technology 3(2) (1995) 245–248.

[45] W. K. Ho, C. C. Hang, L. S. Cao, Tuning of pid controllers based on gain and phase margin specifications, Automatica 31(3) (1995) 497–502.

[46] M. Rossi, C. Scali, Monitoraggio delle prestazioni di processi controllati: Sviluppo di un sistema per individuare anomalie di funzionamento, Atti Convegno GRICU 2004 ”Nuove Frontiere di Applicazione delle Metodologie dell’Ingegneria Chimica”; Ischia (I) II (2004) 1355–1358.

[47] C. Scali, M. Rossi, An automatic system for closed loop performance monitoring, Proceedings of IcheaP-7, 7th Int. Conf. on Chemical and Process Engineering, Chem. Eng. Transactions, ISBN 88-900775-7-3 6 (2005) 155–160.

[48] M. Rossi, A. K. Tangirala, S. L. Shah, C. Scali, A data-base measure for interactions in multivariate systems, ADCHEM-2006 Int. Conf.: Advanced Control of Chemical Processes (2006) 681–686.

[49] K. A. Hoo, M. J. Piovoso, P. D. Schnelle, D. A. Rowan, Process and controller performance monitoring: overview with industrial applications, International Journal of Adaptive Control and Signal Processing 17 (2003) 635–662.

[50] W. L. Bialkowski, Dreams versus reality: a view from both sides of the gap, Pulp and Paper Canada 94 (1993) 19–27.

[51] D. J. Kozub, Controller performance monitoring and diagnosis: experiences and challenges, In: AIChE Symposium Series 94 (1997) 83–96.

[52] M. A. M. Persechini, A. E. C. Peres, F. G. Jota, Control strategy for a column flotation process, Control Eng. Practice 12(8) (2004) 963–976.

[53] D. Chen, D. E. Seborg, Relative gain array analysis for uncertain process models, AIChE Journal 48(2) (2002) 302–310.

[54] M. Daszykowski, B. Walczak, D. L. Massart, Looking for natural patterns in data. part 1: Density-based approach, Chemometrics and Intelligent Laboratory System 56 (2001) 83–92.

[55] B. Huang, S. L. Shah, Performance Assessment of Control Loops, Theory and Applications, Springer Verlag, London, 1999.

[56] T. E. Marlin, Process Control: Designing Processes and Control Systems for Dynamic Performance, McGraw Hill, Boston, 2000.

[57] R. K. Wood, M. W. Berry, Terminal composition control of a binary distillation column, Chem. Eng. Sci. 29.

[58] S. C. Patwardhan, S. L. Shah, From data to diagnosis and control using generalized orthonormal basis filters. part 1: Development of state observers, J. of Process Control 15(7) (2005) 819–835.

[59] S. C. Patwardhan, S. Manuja, S. Narasimhan, S. L. Shah, From data to diagnosis and control using generalized orthonormal basis filters. part 2: Model predictive and fault tolerant control, J. of Process Control 16(2) (2004) 157–175.

[60] K. H. Johansson, The quadruple tank process: a multivariable laboratory process with an adjustable zero, IEEE Trans. on Control System Technology 8(3) (2000) 456–465.