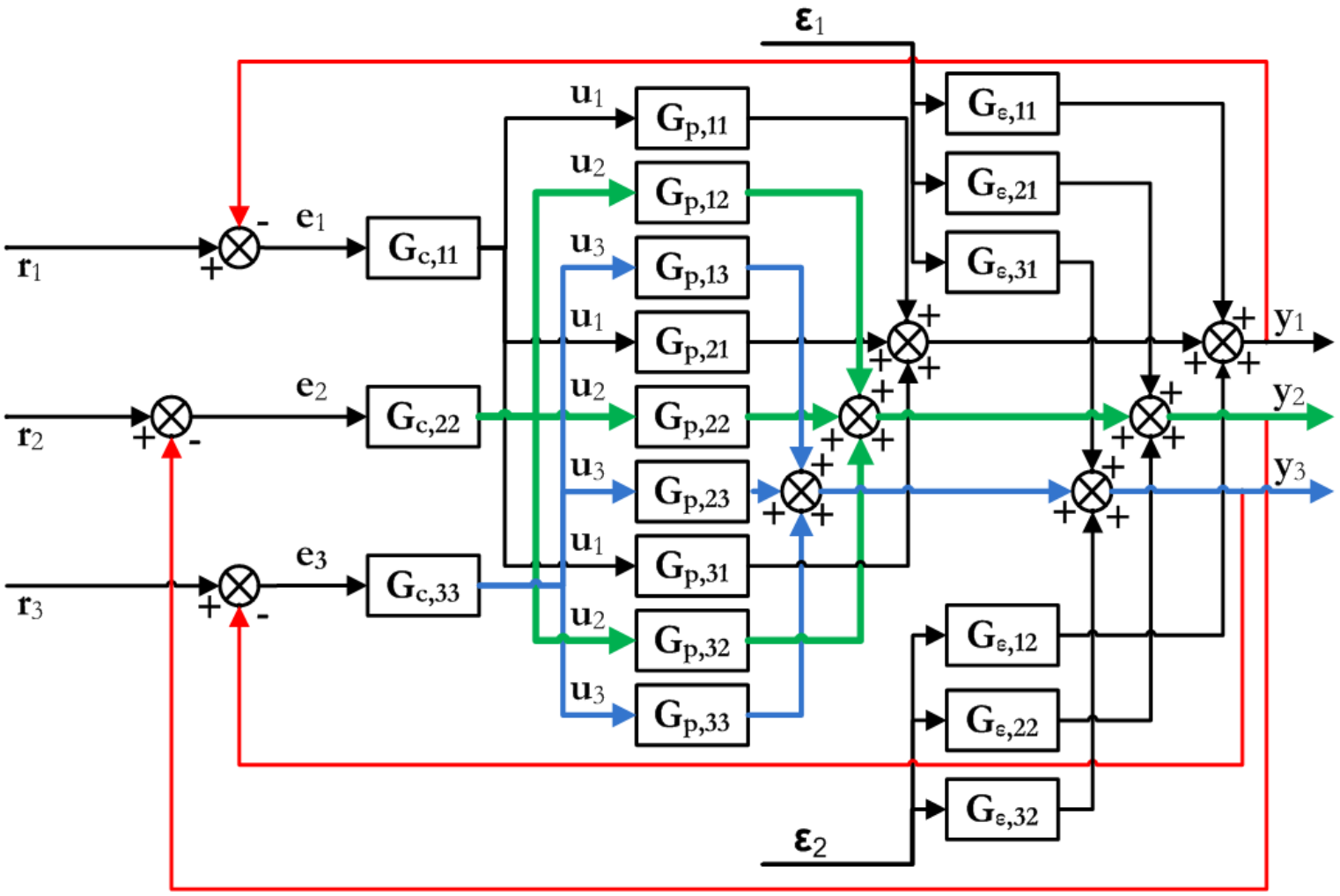


Rui Telmo Ferreira dos Reis

Multivariable controller performance monitoring

Master Thesis in the scientific area of Chemical Engineering submitted to the Department of Chemical Engineering, Faculty of Science and Technology, University of Coimbra

February - 2014

· U C ·

UNIVERSIDADE DE COIMBRA

· U C · FCTUC FACULDADE DE CIÊNCIAS E TECNOLOGIA UNIVERSIDADE DE COIMBRA

Master Thesis

Rui Telmo Ferreira Reis

Multivariable controller performance monitoring

Master Thesis in the scientific area of Chemical Engineering, supervised by Doctor Natércia Fernandes and Doutor Andrey Romanenko submitted to the Department of Chemical Engineering, Faculty of Science and Technology, University of Coimbra

Supervisors:
Doctor Natércia Fernandes
Doctor Andrey Romanenko

Financiamento/ Financing:

NAMPI project – Reference 23 007
NEW APPLICATIONS FOR INDUSTRIAL PROCESS MONITORING
QREN via Mais Centro regional program
European Union via FEDER framework program

Coimbra
2014

· U C ·

UNIVERSIDADE DE COIMBRA

Contents

Acknowledgments

Gostaria de agradecer à minha orientadora Doutora Natércia Fernandes pela oportunidade que me deu e pela sua dedicação na transmissão de conhecimentos e na elaboração desta tese. O muito obrigado e peço desculpa por todo o outro trabalho que tinha para fazer e que ficou para trás.

Ao Doutor Andrey Romanenko pela atenção e pelo trabalho que teve na minha tese.

À Mestre Ana Brásio por partilhar muita informação que depois usei nesta tese.

Ao Doutor Lino Santos pelo apoio dado.

Ao Doutor Ermelindo Leal pela ajuda, pelo apoio e pela companhia que me deu durante a elaboração deste trabalho.

Ao Mestre Paulo Dias pelo apoio e pela companhia que me deu durante o trabalho.

Aos meus Pais e à minha Irmã por todo o apoio que me deram desde sempre.

Abstract

The general purpose of this work is to assess and monitor the performance of MIMO control systems. It is divided into two parts. In the first part, a software tool that assesses the performance of a control system through a set of industrial data was developed. This tool assesses the performance comparing the variability of the control system in normal circumstances with the variability of the same process under the action of a minimum variance controller. The tool was developed taking into account that it should work with the least possible information of the process. In the second part of this work, existing methods to prioritize the maintenance of control loops were analysed comparatively, based on the level of interaction between control loops.

This work presents three procedures/programs (in `octave`) to calculate the interactor matrix needed to assess the performance of a control system based on the minimum variance controller.

To establish the priority of the control loops for maintenance, the following methods were tested: the modified IAE technique from Rossi et al. (2006), the canonical correlation method from Rahman and Choudhury (2010, 2011), the variability matrix from Farenzena et al. (2009), the *LoopRank* method from Farenzena and Trierweiler (2009) and the methods using the normal IAE and ISE from Rahman and Choudhury (2010, 2011).

These methods were implemented in `Octave` and tested using different datasets of generated data. The interactor matrix is better estimated from routine data with a larger dither signal. The results show that the performance of a control system is lower when the interactions between control loops are stronger. A considerable number of the methods tested to prioritize control loops was not able to predict the expected loop rank for all the tests. Only the methods which used the variability matrix and the error values (IAE and ISE) worked consistently through all the tests.

A possible future work is to extend the potentialities of the developed software by incorporating other benchmarks (LQG, MPC).

In what concerns the loop prioritization for maintenance, the creation of a method that included information from the manipulated variables seems to be a possibility that is worth to be exploited.

Resumo

O objectivo principal deste trabalho é avaliar e monitorizar o desempenho de um sistema de controlo MIMO. Este trabalho foi dividido em duas partes. Numa primeira parte foi desenvolvido uma ferramenta informática que avalia o desempenho de um sistema de controlo através de dados industriais. Para avaliar o desempenho, esta ferramenta compara a variabilidade do sistema de controlo real com a variabilidade desse mesmo sistema sujeito à acção de um controlador de variância mínima. No seu desenvolvimento teve-se em conta que deve tentar evitar requerer o conhecimento explícito dos modelos processuais. Na segunda parte deste trabalho, analisam-se comparativamente métodos existentes para definir a prioridade de acções de manutenção nos vários ciclos de controlo, com base no estudo do nível de interacções entre eles.

Este trabalho apresenta três procedimentos/ programas (em `Octave`) para determinar a matriz de interacção necessária para avaliar o desempenho do sistema de controlo, utilizando como referência o controlador de variância mínima.

Para definir as prioridades das acções de manutenção dos vários ciclos de controlo foram testados os seguintes métodos: técnica do IAE modificado de Rossi et al. (2006), método da análise da correlação canónica de Rahman and Choudhury (2010, 2011), matriz de variabilidade de Farenzena et al. (2009), método do *LoopRank* de Farenzena and Trierweiler (2009) e métodos que utilizam o IAE e o ISE de Rahman and Choudhury (2010, 2011). Estes métodos foram implementados em `Octave` e testados em vários conjuntos de dados gerados por simulação. Verificou-se que a estimativa da matriz de interacção através dos dados de operação é melhor quando o sinal de ruído introduzido no processo (dither signal) é maior. Observou-se ainda que as interacções entre os ciclos fazem diminuir o desempenho do sistema de controlo.

Um número considerável dos métodos testados para definir a prioridade da manutenção não sugeriu, para a totalidade dos testes, a ordem que à partida seria expectável em cada caso. Apenas os métodos que utilizam a matriz de variabilidade e os valores dos erro (IAE e ISE) levaram a resultados consistentes tendo em atenção a totalidade dos testes.

Um eventual caminho a seguir, em trabalho futuro, é estender as capacidades do software

desenvolvido incorporando a comparação com o desempenho de outros controladores óptimos (LQG, MPC).

Relativamente à definição de prioridades na manutenção dos ciclos de controlo, a criação de um novo método que incluísse informação proveniente das variáveis manipuladas afigura-se uma possibilidade que valerá a pena ser explorada.

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

The work presented in this thesis was carried out in the context of NAMPI, a collaborative research and development project between CIENGIS and the University of Coimbra, in the area of process monitoring of industrial systems.

This dissertation focuses on the development and assessment of tools for assessing and monitoring the performance of the multivariate control systems (MIMO). These tools must be easy to use, robust, based on data obtained easily from the process and should be as less invasive as possible. Performance assessment of univariate control systems is an area that has begun to develop in large scale in 1989 (Harris, 1989) and had its peak in the decades of 1990 and 2000. Currently, the research on performance assessment is more focused on MIMO control systems and this work is a contribution in this area.

A possible approach of performance assessment of MIMO control systems is via the comparison of real data with data produced by the system considered to be in optimal conditions, i.e., with a benchmark. In this work, the optimal conditions is to use a minimum variance controller. This type of controller is not ideal for use in many processes, provokes more wear on the devices and greater variability in the control action, however, is very useful as a performance criterion. The determination of the interactor matrix is one of the most important, most sensitive and most complicated step in assessing the performance of MIMO systems.

One of the reasons for performance degradation of MIMO control system is the interaction among control loops. Therefore, it is important to detect, diagnose, quantify loop interactions and rank loops for maintenance scheduling. This work presents several recent data based methods to identify control loop interactions and maintenance prioritization. This work compares

the performance of these methods by simulation using the data presented by Rossi et al. (2006).

1.2 General framework

The main purpose of control systems is to maximize profits by transforming raw materials into products while fulfilling a set of conditions (e.g. product quality specifications, operational constraints, safety and environmental norms). The design, tuning and implementation of control strategies and controllers are assumed within the first phase of process control. When comissioned, the control system must be working well, however, after some time the performance may degrade. Therefore, the second phase of process control should be the supervision of control loops and the early detection of performance deterioration. This task is made mainly by maintenance and control staff within the framework of control performance monitoring (CPM).

Today, the process industries are faced with ever-increasing demands on product quality, productivity and environmental regulations. Therefore, these companies want to operate their plants at top performance, with control systems with consistently high performance (Jelali, 2010).

There are many studies that show the extent of problem of the weak performance of countless controllers in large industrial companies (see Paulonis and Cox, 2003; Ruel, 2003 and Torres et al., 2006).

Some causes for poor controller performance are: changing operating conditions without making new tuning, variations in the state of the plant equipment, such as wear, increased friction, and failures in software or hardware. There are reports of cases where the controller has never been tuned and retains the original manufacturer's tuning (Hovd, 2009). Some equipment in the control loop may require maintenance or replacement (e.g. faulty measurements, control valves with excessive stiction, severe fouling in heat exchangers) (Hovd, 2009). Another cause of performance degradation is due to the increasing interaction between the control loops due to the increasing efficiency of the whole plant. For example, the energy integration of various control loops optimizes energy consumption, but increases the interaction among control loops. In this case, to maintain closed-loop performance of the process, it may be necessary to use an advanced controller.

1.3 Synopsis

Chapter 1 presents the motivation and the general framework of this work. Besides, a brief description of some relevant concepts is carried out.

Chapter 2 presents the state of the art with a brief literature review of the performance assessment of control systems and of methods for determining the interactor matrix. At the end, various approaches are presented to assess and prioritize control loop maintenance.

In chapter 3, software prototypes to calculate the interactor matrix and the actual performance of control systems are presented. These prototypes are tested with simulated examples and the influence of certain factors is studied in the determination of the interactor matrix and in the calculation of process performance.

Methods for the evaluation of interactions and prioritization of control loops are presented in chapter 4. These methods are tested and compared with the same process.

The dissertation ends with conclusions and an indication of future work.

1.4 Revision of concepts

A typical feedback control configuration is given in Figure 1.1. This is a system with a process, a controller, sensors and actuators. The *regulator problem* is to manipulate u (manipulated variable or control signal) to neutralize the effect of a disturbance ε. The *servo problem* is to manipulate u to keep the output y (output or controlled variable) close to a given reference input r (set-point). Thus, in both cases the control error, $e(t) = -y(t) + r(t)$, must be minimized.

The role of controller may be to stabilize open-loop unstable plants, to improve the efficiency and the robustness of the process and to remove the effects of disturbances.

A general open-loop linear process model structure is given by

$$y = \mathrm{G_p} u + \mathrm{G_\varepsilon} \varepsilon, \tag{1.1}$$

where $\mathrm{G_p}$ represent the process transfer function and $\mathrm{G_\varepsilon}$ represent the disturbance transfer function.

In Figure 1.1 is shown a control system in closed-loop, where $\mathrm{G_c}$ represents the controller transfer function while $\mathrm{G_m}$ is the sensor transfer function. This system is said to be a closed-loop system since the parts are interconnected in a cycle.

The input to the system may consist of two types of variables: disturbance variable $\varepsilon(t)$ and

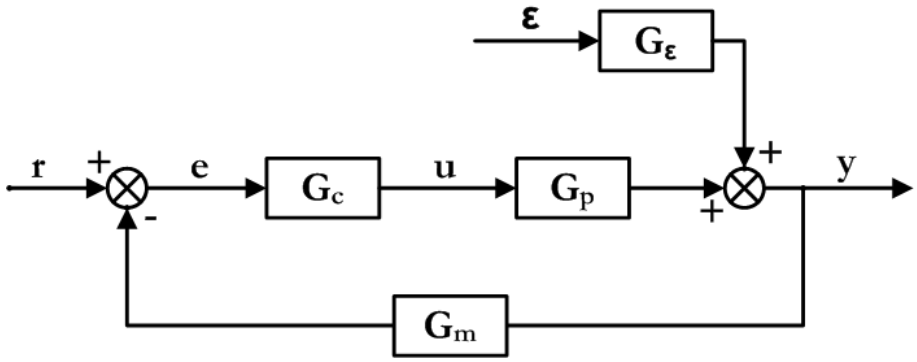


Figure 1.1: A schematic of feedback control loop.

manipulated variable ($u(t)$). The former is typically external to a system out of control of the operator. The manipulated variable is controlled by the controller and it drives the system in a desired way.

The **feedback control** provide robustness to uncertainty, measuring the difference between the observed value of the output variable and its desired value. When the system undergoes some change that affects the output signal, the controller tries to force the system back to the desired operating point.

Since feedback is reactive, there must be an error before corrective actions are taken. However, in some circumstances it is possible to measure a disturbance and this information may be used to take corrective action before the disturbance has influenced the system. This way, the effect of the disturbance is thus reduced by generating a control signal that counteracts it. Such control configuration is called **feed-forward control**.

When analysing a set-point response, the criteria used to describe how well the process responds to the change can include the **rise time, settling time, decay ratio, overshoot and steady-state offset**.

1. *Rise time* (t_r): The time it takes for the output to first reach 90 % of its final value.

2. *Settling time* (t_s): The time after which the output remains within $\pm 5\%$ of its final value.

3. *Overshoot*: The peak value divided by the final value, which should typically be $1.2(20\%)$ or less.

4. *Decay ratio*: The ratio of the second and first peaks, which should typically be 0.3 or less.

5. *Steady-state offset*: The difference between the final value and the desired final value, which is usually required to be small.

The rise time and settling time are measures of the speed of the response, whereas the overshoot, decay ratio and steady-state offset are related to the quality of the response.

1.4.1 Interactions

In the industry, each control loop can have many different disturbances that influence its performance. The disturbance of a control loop can be done through the **noise** generated by the equipment / devices connected to the process, through changes in the set-point or by other control loops with significant interactions.

A noise signal is a signal produced by a stochastic process and it is random (unpredictable). In this work, only **white noise** will be considered. This noise is a sequence of an uncorrelated random variable with zero mean and finite variance. This noise is defined by adjusting only the variance of the signal, σ_ε^2.

To know if the disturbance is due to other control loops, it is necessary to analyze the interactions among control loops. An initial analysis can be done via a **cross-correlation analysis**. This correlation can be described as the linear relationship between two variables. The cross-correlation $\mathrm{R}(i)$ is defined (Rice et al., 2010)

$$\mathrm{R}_{\mathrm{v,p}}(i) = \frac{\sum_i [(y_\mathrm{v}(i) - \bar{y}_\mathrm{v})(y_\mathrm{p}(i) - \bar{y}_\mathrm{p})]}{\sqrt{\sum_i (y_\mathrm{v}(i) - \bar{y}_\mathrm{v})^2}\sqrt{\sum_i (y_\mathrm{p}(i) - \bar{y}_\mathrm{p})^2}}, \tag{1.2}$$

where y_v and y_p are the output variables of the control loops v and p. $\bar{y}_\mathrm{v}$ is the average value of the output variable of the control loop v.

There is a direct connection if the correlation value is positive and an inverse relationship if the value is negative. If there is no significant relation between the two control loops, the correlation value remains close to zero.

In the case where the inputs and outputs are correlated, a better measure of the interaction is the **partial correlation**. Considering the process with three control-loops (loops v, p and q), the partial correlation between the output variables of the loops v and p is a measure of the relationship between these two variables, eliminating the effect of the output variable of the loop q. The partial correlation between the loops v and p ($\rho_{\mathrm{v,p|q}}$) is

$$\rho_{\mathrm{v,p|q}} = \frac{\mathrm{R}_{\mathrm{v,p}} - \mathrm{R}_{\mathrm{v,q}}\mathrm{R}_{\mathrm{p,q}}}{\sqrt{(1 - \mathrm{R}_{\mathrm{v,q}}^2)(1 - \mathrm{R}_{\mathrm{p,q}}^2)}}. \tag{1.3}$$

Auto-correlation, AC is a mathematical tool for finding repeating patterns, such as the

presence of a periodic signal or for identifying the missing fundamental frequency in a signal. It describes the correlation between values of the process variable at different times ($t = i$ and $t = k+1$) (Rice et al., 2010)

$$\mathrm{AC}(k) = \frac{\sum_i [(y(i) - \bar{y})(y(i-k) - \bar{y})]}{\sum_i (y(i) - \bar{y})^2}, \tag{1.4}$$

where y is the output variable and $\bar{y}$ is the set-point or the series average if there is an offset. k is the time delay in samples. The AC value must lie in the range $[-1, 1]$. If data is random, the values will be approximately zero for all time intervals.

A fundamental test for assessing the performance of control loops is to check the auto-correlation of the output samples: the autocorrelation should die out beyond the time delay, which is a feature of the minimum variance (MV) controller. If there are some significant auto-correlation values that persist beyond the time delay of process, the control performance deviates substantially from the MV performance bound (Jelali, 2010).

The Canonical correlation is a technique for analyzing the relationship between two sets of variables, where each set may contain several variables.

Considering two sets of variables that must be evaluated, a set called $\mathrm{X_{cc}}$ with p variables $(x_1, x_2, \cdots, x_p)$ and another called $\mathrm{Z_{cc}}$ with q variables $(z_1, z_2, \cdots, z_q)$, the canonical correlation analysis calculates a new pair of the variables (w_1, v_1) that are a linear combination of a set of original variables ($\mathrm{X_{cc}}$,$\mathrm{Z_{cc}}$). These new variables are called the scores or the canonical variates

$$\mathrm{w_1} = \mathrm{a_{1,cc}x_1} + \mathrm{a_{2,cc}x_2} + \cdots + \mathrm{a_{p,cc}x_p}, \tag{1.5}$$

$$\mathrm{v_1} = \mathrm{b_{1,cc}z_1} + \mathrm{b_{2,cc}z_2} + \cdots + \mathrm{b_q z_{q,cc}}. \tag{1.6}$$

The new variables are calculated in a way that the cross-correlation between the two canonical variables is maximized.

This correlation between the two canonical variables is the first canonical correlation. The coefficients of the linear combination $(\mathrm{a}_{i,\mathrm{cc}}, \mathrm{b}_{i,\mathrm{cc}})$ are canonical coefficients or canonical weights.

The first canonical correlation represents the highest possible correlation between any linear combination of the two sets of variables. The canonical correlation analysis will search a second set of canonical variables, uncorrelated with the first pair, that produces the second highest correlation coefficient. The process of constructing canonical variables continues until the number of pairs of canonical variables equals the number of variables in the smaller group.

1.4.2 Model Structures

The discrete time transfer functions may be identified using time series statistical methods.

The most general model is called **PEM model (Prediction Error Model (PEM)** and it is described by (1.7).

$$\mathrm{A}(z^{-1})\mathrm{y} = \frac{\mathrm{B}(z^{-1})}{\mathrm{F}(z^{-1})}\mathrm{u} + \frac{\mathrm{C}(z^{-1})}{\mathrm{D}(z^{-1})}\varepsilon \tag{1.7}$$

Where A, B, C, D and F are polynomials functions. Several methods have been developed for the identification of prediction error models. Some of these methods use regression-based techniques and are based on the premise that models are primarily used for predicting future trends. These methods construct a one-step ahead predictor for the selected model and seek to minimize the error between the obtained data and this predictor. If the residuals can be shown to be white noise, then the resulting model is probably good.

Depending on which polynomials have a nonzero order, different types of models can be obtained.

The **ARX model (AutoRegressive with eXogenous Input Model)** has $\mathrm{C}(z^{-1}) = \mathrm{D}(z^{-1}) = \mathrm{F}(z^{-1}) = 1$.

The **ARMAX model (AutoRegressive Moving Average with eXogenous input)** has $\mathrm{D}(z^{-1}) = \mathrm{F}(z^{-1}) = 1$.

The **BJ model (Box-Jenkins Model)** has $\mathrm{A}(z^{-1}) = 1$.

The **OE model (Output Error Model)** has $\mathrm{A}(z^{-1}) = \mathrm{C}(z^{-1}) = \mathrm{D}(z^{-1}) = 1$.

The **FIR model (Finite Impulse Response Model)** has $\mathrm{A}(z^{-1}) = \mathrm{C}(z^{-1}) = \mathrm{F}(z^{-1}) = \mathrm{D}(z^{-1}) = 1$.

The **ARMA model (AutoRegressive Moving Average Model)** has $\mathrm{B}(z^{-1}) = \mathrm{F}(z^{-1}) = 0$ and $\mathrm{D}(z^{-1}) = 1$.

The **MA model (Moving Average Model)** has $\mathrm{B}(z^{-1}) = \mathrm{F}(z^{-1}) = 0$ and $\mathrm{A}(z^{-1}) = \mathrm{D}(z^{-1}) = 1$.

The **AR model (AutoRegressive Model)** has $\mathrm{B}(z^{-1}) = \mathrm{F}(z^{-1}) = 0$ and $\mathrm{C}(z^{-1}) = \mathrm{D}(z^{-1}) = 1$.

The multivariate form of the ARMA model is called **ARMAV model (AutoRegressive Moving Average Vector or Vector ARMA process)**. The ARMAV (p,q) model for a stationary multivariate time series has the following form.

$$\mathrm{Y}(:,t) = \phi_{\mathrm{AR,1}}\mathrm{Y}(:,t-1) + \cdots + \phi_{\mathrm{AR,p}}\mathrm{Y}(:,t-p) + \varepsilon(:,t) - \phi_{\mathrm{MA,1}}\varepsilon(:,t-1) - \cdots - \phi_{\mathrm{MA,q}}\varepsilon(:,t-q) \tag{1.8}$$

For each instant, $\mathrm{Y}(:,t)$ is a vector of values, where each value is one of the $\mathrm{n_Y}$ output variables. The $\varepsilon(:,t)$ represents a multivariate white noise. The $\phi_{\mathrm{AR},i}$ and $\phi_{\mathrm{MA},i}$ are $\mathrm{n_Y} \times \mathrm{n_Y}$ matrices with autoregressive (AR) and moving average (MA) parameters, respectively.

Chapter 2

State of the Art

2.1 Process control performance assessment

The development of control performance assessment technology began in 1989 by Harris (1989), who demonstrated that the minimum variance benchmark can be estimated from normal closed-loop operation data. He created the **Harris index** that compares real performance with the performance of a minimum variance controller, for SISO systems. This method is theoretically sound, efficient, reliable, readily interpretable and computationally simple. To determine this index, it is only necessary to know the process time delay and to have closed loop output data. If the values of the autocorrelation function of the routine closed-loop data are statistically significant beyond the delay of the process, the controller is not a minimum variance controller (Rice et al., 2006). The theory of minimum variance control was created by Aström (1970) and Box et al. (1970). DeVries and Wu (1978) used these ideas for performance assessment to analyze the effectiveness of basis-weight control of a paper machine. Another related performance assessment index defined as the normalized performance index was proposed by Desborough and Harris (1992).

The concept of MV benchmarking was extended to feedback/feedforward loops by Desborough and Harris (1993) and by Stanfelj et al. (1993). The performance assessment for cascade control systems was developed by Ko and Edgar (2000).

Minimum variance benchmarking considers only the most fundamental performance limitation of a control loop, which is the existence of time delays. In practice, however, there are many other limitations on the achievable control performance, such as constraints on controller order, structure and action. A controller showing poor performance compared to MVC is not necessarily a bad controller. This controller may have greater variability of the controlled

variable and a lower variability of control action.

Extensions of the Harris index to unstable and non-minimum-phase systems have been reported by Tyler and Morari (1995) and Tyler and Morari (1996). In 1996, Lynch and Dumont used Laguerre networks to evaluate the performance index. Several modified versions of the Harris index were introduced including design specifications of the user, leading to more realistic performance indices (Kozub and Garcia, 1993) and (Horch and Isaksson, 1999).

Many authors decided to use historical data to form a benchmark, a period of "golden" operation data when a desirable control performance was achieved (Huang, 1997) and (Patwardhan et al., 1998).

Swanda and Seborg (1997) and Swanda and Seborg (1999) proposed the dimensionless settling time of the closed-loop and the dimensionless integral of absolute value of control error as performance indices.

A large percentage of the implemented controllers in industry are a kind of PID controllers that have order, structure and action constraints. Therefore, realistic benchmarks should be applied for their assessment, such as those proposed by Eriksson and Isaksson (1994) and by Ko and Edgar (1998).

Huang (1997) proposed the Linear-Quadratic Gaussian (LQR) as an alternative to or the next step after applying the MV benchmark, when the latter indicates poor performance. This benchmark takes into account the control effort in the performance assessment.

Grimble (2002) suggested the Generalized Minimum Variance (GMV) benchmark as a simplified LQR benchmark.

The control performance monitoring techniques for time-varying systems was proposed by Huang (2002) and by Olaleye et al. (2004). Majecki and Grimble (2004a) studied this type of techniques for non-linear systems. All developments described above were made for SISO systems.

2.2 MIMO control performance assessment

The control performance assessment of SISO systems is attractive by its simplicity, however, it has to be used on the loop-by-loop basis, ignoring the interactions between control loops. A typical industrial process contains at least a few interacting control loops and it is desirable to assess their performance as a whole. Therefore, the MIMO controller benchmark is required.

DeVries and Wu (1978) used multivariate time series techniques to analyse the effectiveness of basis-weight control of a paper machine. Process control effectiveness is measured by

comparing the observed output variation with an estimate of the theoretical minimum variation obtained from autoregressive moving-average vector (ARMAV) time-series models.

In the late 90s, the performance assessment with minimum variance (MV) benchmark was extended to MIMO control systems (Harris et al., 1996; Huang et al., 1997a; Huang et al., 1997b and Huang and Shah, 1998). Huang et al. (1997a) developed a filtering and correlation analysis algorithm (FCOR) for MIMO feedback control performance assessment as well as the minimum variance benchmark and the singular linear quadratic (LQ) benchmark. The singular LQ control is the LQ control without penalty of the control action. Later, McNabb and Qin (2003) and McNabb and Qin (2005) developed a method in state space to determine the MV benchmark for MIMO control systems.

Harris et al. (1996) developed a parallel approach based on a spectral interactor and spectral factorization, which can also extract the minimum variance term from routine operating data.

In the performance assessment of MIMO feedback systems, the interactor matrix is a very important part in the calculation of the performance index. It is one of the most computationally demanding operation. Therefore, there have been many researchers that attempted to develop alternative methods that do not require this matrix.

The concept of MV benchmarking was extended to feedback/feedforward loops by Huang et al. (2000).

As an alternative, Jelali (2010) proposed to calculate a lower bound and a upper bound of the MIMO performance index. Based on works of Ettaleb (1999), and of Ko and Edgar (2001b), Huang et al. (2005b) and Xia et al. (2006), he proposed a practical method and procedure to assess the performance of the multivariable control systems, requiring only closed-loop data from normal process operation and the knowledge of time delays.

The research direction of Qin and Yu (2007), Yu and Qin (2008a) and Yu and Qin (2008b) is to create a benchmark from historical data.

A multivariable version of the GMV control assessment was derived by (Majecki and Grimble, 2004b) using the concept of the interactor matrix of the generalized plant.

Currently, a large part of the multivariable control systems are constituted by model predictive controllers and to assess them there are many works that detail specific MPC monitoring methods (Patwardhan et al., 1998), (Patwardhan, 1999), (Ko and Edgar, 2001a), (Schäfer and Çinar, 2002), (Schafer and Cinar, 2004), (Gao et al., 2003) and (Julien et al., 2004).

2.2.1 Interactor Matrix

The concept of the interactor matrix was introduced as a multivariable generalization of the SISO time delay term. The performance of a minimum variance controller is obtained with the delay value of the process for SISO systems. The delay term is important to design the MIMO minimum variance controller. The interactor matrix was first proposed by (Wolovich and Falb, 1976) and had a lower triangular form. With this form, the minimum variance control law depends on the order of output variables, and in this case, it is common to place first the most important variables.

The interactor matrix is defined for any proper, rational polynomial transfer function of process G_p (r x m matrix) as the *unique*, non-singular (r x r) *lower left triangular* polynomial matrix D which satisfies the following conditions:

$$\begin{aligned} \det(D(q)) &= q^n \\ \lim_{q^{-1}\to 0} D(q)G_p(q) &= K \\ \lim_{q^{-1}\to 0} \tilde{G}_p(q) &= K, \end{aligned} \tag{2.1}$$

where K is a finite and full rank matrix, n is the number of infinite zeros of G_p and $\tilde{G}_p$ is the delay free transfer function matrix of G_p which contains only finite zeros. The matrix D is defined as the **interactor matrix** and can be written in the Markov parameter representation

$$D = D_0 q^{\theta} + D_1 q^{\theta-1} + \cdots + D_{\theta-1} q^1, \tag{2.2}$$

where θ is the **order of the interactor matrix**, D is unique for a given transfer function matrix, and $D_i (i = 0, 1, \cdots, \theta - 1)$ are matrices with coefficients.

Rogozinski et al. (1987) proposed an algorithm for factorization of the nilpotent interactor matrix which has the full matrix form. Peng and Kinnaert (1992) found the existence of the **unitary interactor matrix**, which is a special form of the nilpotent interactor matrix. In this unitary matrix formulation, the minimum variance control law does not depend on the order of output variables (Huang, 1997). If an interactor matrix fulfill $D^T(q^{-1}) \times D(q) = I$, then it is designated by **unitary interactor matrix**.

Only the unitary interactor matrix is desirable for multivariate performance assessment using minimum variance control as the benchmark.

The minimum variance control law minimizes the variance of the interactor-filtered variable

$\tilde{Y}$, where $\tilde{Y} = q^{-\theta} D Y$. Therefore, all methods / algorithms that use the minimum variance benchmark in MIMO systems, begin by filtering the data with the interactor matrix.

As mentioned previously, for a given transfer function matrix of a MIMO process there is not a unique unitary interactor matrix. Two interactor matrices, D_1 and D_2 are "equivalent" if $D_1 = \Gamma D_2$ where Γ is an r x r, unitary real matrix ($\Gamma^T \Gamma = I$) (Huang, 1997).

There is another interactor matrix that it is not unitary and that satisfies the following condition

$$D_w^T(q^{-1}) D_w(q) = W, \tag{2.3}$$

where W is greater than zero and it is a symmetric weighting matrix. This matrix, D_w is designated as the **weighted unitary interactor matrix**. An example of the W matrix for a control system with two output variables is

$$W = \begin{bmatrix} 1 & 0 \\ 0 & 4 \end{bmatrix}, \tag{2.4}$$

where the second output variable is considered more important, the control law obtained will further minimize the variation of the second output variable. The weighted unitary interactor matrix can be obtained by the following equation

$$D_w(q) = D(q) W^{1/2} = \Gamma D(q) W^{1/2}, \tag{2.5}$$

where $D(q)$ is a unitary interactor matrix.

2.2.2 Methods of Interactor Matrix calculation

When the process transfer function G_p is known, which is a strong requirement in the CPM context, a unitary interactor matrix can be determined using an algorithm developed by Rogozinski et al. (1987) and by Peng and Kinnaert (1992).

There are two factorizations of the transfer functions that are used in control and one of them is used in the Rogozinski's algorithm. This factorization is called Right Matrix Fraction Description (RMFD) and can be summarized as (Goodwin et al., 2001)

$$G_p = N[R]^{-1}, \tag{2.6}$$

where R is the matrix with the polynomials of the denominators of the G_p matrix and N is the

matrix with the polynomials of the numerators of the G_p matrix.

Rogozinski's algorithm evaluates a nilpotent interactor matrix from the matrix of coefficients of the numerator of a right matrix fraction description (RMFD) of the system. A brief description of the method based on the work of Rogozinski et al. (1987) and Peng and Kinnaert (1992) is described in Appendix B.

The interactor matrix, if the dynamic model of the multivariable process is not available, may be obtained using the Markov parameters. This method has two options, and the first one is to obtain the Markov parameters directly from a time series model (e.g. a first-order autoregressive model). The second option is to obtain these parameters directly through the data routine of the process (Huang, 1997 and Jelali, 2010).

Although it is possible to obtain the interactor matrix from routine data, this option has some disadvantages. For example, sometimes it is not be possible to obtain the interactor matrix due to computational problems or information poor data. The other disadvantage is the need to introduce extra noise (**Dither Signal**) to the process to stimulate and thus be able to properly calculate the Markov parameters of the process.

These two approaches for the interactor matrix determination are summarized in Appendix E.

2.2.3 MIMO Feedback Controller Performance Assessment

Huang et al. (1997a) presented an efficient, stable filtering and correlation (FCOR) method to estimate the minimum variance (MV) benchmark for MIMO control systems. A MIMO process can be modeled as

$$\mathrm{Y} = \mathrm{G_p U} + \mathrm{G}_\varepsilon \varepsilon, \tag{2.7}$$

where Y is the vector of output variables, G_p is the proper process transfer functions matrix, U is the vector of manipulated variables, G_ε is the proper disturbance transfer functions matrix and ε is the white noise sequence.

According to the law of minimum variance control (see Appendix A), the controller determines the minimum variance with the output variables filtered by the interactor matrix. The interactor-filtered output variable is represented by

$$\tilde{\mathrm{Y}} = q^{-\theta} \mathrm{D\,Y}, \tag{2.8}$$

where θ is the order of the interactor matrix. Huang and Shah (1999) showed that since D is a

unitary interactor matrix, the minimum variance control law that minimizes the objective function of the interactor-filtered variable, $\tilde{\mathrm{Y}}$ also minimizes the objective function of the original variable Y.

The benchmark of minimum variance of a MIMO process is

$$\mathrm{E}(\tilde{\mathrm{Y}}^{\mathrm{T}}\tilde{\mathrm{Y}}) = \mathrm{trace}(\mathrm{cov}_{\mathrm{F}\varepsilon}), \tag{2.9}$$

where *cov* is the covariance matrix and *trace* is the sum of the diagonal elements of a matrix. The minimum variance term $\mathrm{F}\varepsilon$, may be estimated from routine operating data Y. Considering a closed loop process $\mathrm{G}_{\mathrm{p,cl}}$

$$q^{-\theta}\mathrm{D}\,\mathrm{G}_{\mathrm{p,cl}} = \mathrm{F} + q^{-\theta}\mathrm{R}, \tag{2.10}$$

with F equal to the first $(\theta - 1)$ terms of $\tilde{\mathrm{G}}_{\mathrm{p,cl}}$ in the expanded form (impulse response). The variability of F does not depend of the control system (from the controller action). This property is also known as the feedback-control invariance property.

$$\mathrm{F} = \mathrm{F}_0 + \mathrm{F}_1 q^{-1} + \cdots + \mathrm{F}_{\theta-1} q^{-\theta+1} \tag{2.11}$$

For the expression of F, it is necessary to identify a time series model based on process data. Then, F is determined using the impulse response of this model. To calculate the benchmark of minimum variance is necessary to identify the white noise sequence ε.

Although ε is unknown, it can be replaced by the estimated innovations sequence. This can be done by pre-whitening the output variable *Y* through process transfer function of the closed loop without any external disturbance $(\mathrm{G}_{\mathrm{p,cl}})$

$$\varepsilon = {\mathrm{G}_{\mathrm{p,cl}}}^{-1}\mathrm{Y}. \tag{2.12}$$

This reverses the relationship between Y and ε. The model $\mathrm{G}_{\mathrm{p,cl}}$ presented in (2.12) is the same model that was obtained to determine the expression of F.

To evaluate the performance of the control system, it is only necessary to compare the covariance matrix of the real process (Σ_{y}) with the covariance matrix of the minimum variance benchmark (2.9).

Most of the MIMO performance indices compare the sum of variances of each output, the trace of the covariance matrix. This type of index is adequate when all variables are independent, which is not a valid assumption in practice.

There are many control loops with the same diagonal covariance matrix but with completely different behaviors in closed loop. Consider the following two 2×2 covariance matrices with identical traces

$$\Upsilon_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}; \quad \Upsilon_2 = \begin{bmatrix} 1 & 0.99 \\ 0.99 & 1 \end{bmatrix};. \tag{2.13}$$

It is obvious that although the diagonal elements are equal, they each represent very different multivariate behavior. Therefore, it is necessary to use a different approach to relate two covariance matrices, especially to deal with the general case of significant correlated output variables.

An example of such approaches is based on **Principal Component Analysis (PCA)** and eigenvalue decomposition (McNabb and Qin, 2003). This procedure requires two covariances matrices, one relative to the current performance and another of the benchmark performance. The latter can be relative to the minimum variance or to the another period based user-defined benchmark from historical data. If $Y(k)$ follows a multivariate normal distribution, $\mathrm{cov}(Y(k))$ is defined as an ellipsoid, whose volume gives an indication of the quality of control (variability of the controlled variable). It is of interest to know how much of this volume of variability in the principal component subspace (PCS) can be reduced by comparing it to the MVC achievable variability. The eigenvalue decomposition (McNabb and Qin, 2003) and (Qin and Yu, 2007) can be performed as

$$\mathrm{cov}(Y(k)) = \mathrm{PSP}^{\mathrm{T}}, \qquad \mathrm{cov}(Y_{\mathrm{mv}}(k)) = \mathrm{P_{mv}S_{mv}P_{mv}^{T}}, \tag{2.14}$$

where $\mathrm{S} = \mathrm{diag}(\lambda_1^2, \cdots, \lambda_{\mathrm{p}}^2)$ and $\mathrm{S_{mv}} = \mathrm{diag}(\lambda_{1,\mathrm{mv}}^2, \cdots, \lambda_{\mathrm{p,mv}}^2)$ in descending order. P is the matrix with eigenvectors.

This new approach uses as a performance index ($\mathrm{I_v}$) the volume ratio of the ellipsoids (McNabb and Qin, 2003)

$$\mathrm{I_v}(l) = \frac{\prod_{i=1}^{l} \lambda_{i,\mathrm{mv}}}{\prod_{i=1}^{l} \lambda_i}. \tag{2.15}$$

where l is the number of principal components. If $\mathrm{I_v}(l)$ is close to 1, then the current control is close to optimal (minimum variance). If $\mathrm{I_v}(l)$ is much lower than 1, then there is a great potential for control performance improvement.

Yu et al. (2008) said that the following equation represents exactly the same index ($\mathrm{I_v}$)

$$\mathrm{I_v} = \frac{\det[\mathrm{cov}(\mathrm{y_{mv}}(k))]}{\det[\mathrm{cov}(\mathrm{y}(k))]}. \tag{2.16}$$

Performance Assessment without the Knowledge of the Interactor Matrix It is desirable to assess the controller performance based on information which is easy to obtain. As was pointed out before, one of the critical step in performance assessment is the determination of the interactor matrix, which is not unique and generally requires the complete knowledge of the entire transfer matrix of the system. Currently, there are some procedures for performance assessment that do not use the interactor matrix. These methods are based on calculation of a lower bound and an upper bound of multivariable minimum variance that can be derived from the knowledge of the order of the interactor matrix alone or from the time delays between inputs and outputs (Jelali, 2010; Huang et al., 2005a and Xia and Howell, 2005).

The lower limit may be determined considering the control system as a sum of independent output variables. The minimum variance of a control loop is always greatest when there is interaction between loops. This comparison is made by considering the smallest delay associated to each output variable ($\theta_{i,\min}$), i.e., for each row in the matrix of transfer functions the value of the smallest delay is considered. The lower bound of overall performance index can be calculated using the procedure described in Section 2.2.3, assuming the interactor matrix as the following diagonal interaction matrix $\mathrm{D} = \mathrm{diag}(\theta_{1,\min}, \cdots, \theta_{p,\min})$.

This limit can be estimated based only on the information of the order of the interactor matrix θ. The upper bound of overall performance index can be calculated using the same procedure, assuming the interactor matrix as,

$$\mathrm{D} = q^{\theta}\mathrm{I}. \tag{2.17}$$

A common assumption is that θ is equal to the largest time delay. Although this bound is not necessarily achievable, it does provide an estimate of the maximum potential to improve the control performance. A procedure presented by Jelali (2010) may be used to determine the order of the interactor matrix.

As mentioned in Section 1.1, the performance assessment consists in a comparison between the performance of the current process and the performance of a benchmark. This can be achieved in different ways. One of these methods assesses the control performance based on a step change in the set-point. Although it is easy to apply, this method has two pratical limitations:

- it requires the deliberate introduction of a process change that is detrimental to process economics (due to increased variability);

- it is more appropriate for processes whose dynamics can be approximated by equations of first or second order, not covering a significant set of processes (Skogestad and Postlethwaite, 2005).

Another type of methods, widely used in industry, compares the actual performance of the current process with the performance of the same process during a previously chosen period taken as benchmark. The main problem of such type of methods is the subjectivity inherent to the choice of the standard time window. The determination of the benchmark period is specially difficult in industrial processes with frequent changes of operating conditions. In light of these limitations, the path followed in this work to evaluate the performance of control loops was to *compare the performance of the closed loop current process with the performance of the process under the action of the minimum variance controller*. In spite of the drawbacks associated with the direct use of this controller in some processes, the value of the performance index obtained with this benchmark is objective since the metric depends only on the process variability.

2.3 Interactions in MIMO systems and loop maintenance prioritization

The assessment of the interaction between control loops is very important in the assessment and monitoring of performance of MIMO control systems. Loop interaction should be avoided because changes in one loop may cause undesirable changes in another loop, deteriorating the overall performance of the MIMO system. Indeed, it is not possible to assume that the variance reduction in one loop will occur without impacting other loops adversely (Farenzena et al., 2009).

There are several methods that use routine data to assess the level of interaction (Rossi et al., 2006; Farenzena and Trierweiler, 2009; Farenzena et al., 2009 and Rahman and Choudhury, 2011).

The number of control loops at plant can be enormous, however, it may be impossible and economically unfeasible to improve and maintain all these loops at their best performance. Therefore, a methodology **to prioritize** loop maintenance is required.

The prioritization of the maintenance actions of control loops can be based on the economic aspect, on the properties of the final product or on the study of the interaction between control loops. When considering only the latter, it favors the maintenance of control loops that interact and affect more other control loops.

Rossi et al. (2006) proposed an easy to use loop prioritization method that requires at least one change of set-point in all control loops. The result of the method is a matrix that shows the relationship between two individual loops. The approach is based on the calculation of the value of the error (IAE modified) of the output variable of the loop where the change of the set-point was done and of another loop. After estimating these two values of IAE, the **Interaction Index, ITD** is calculated.

The obtained results are classified according to Table 2.1 (Rossi et al., 2006). Note that when this index is greater than 0.5, the set-point change in a loop i affect more the other loop j than loop i itself.

Table 2.1: Interpretation of the Interaction Index Rossi et al. (2006).

ITD		Interpretation
[0.000	0.125]	No interaction
[0.125	0.250]	Low interaction
[0.250	0.375]	Medium interaction
[0.375	0.500]	High interaction
[0.500	1.000]	Very High interaction

A limitation of this method is that the results can be affected when the change of set-point coincides with another external disturbance. However, the presence of other set-point changes in the data set can help overcome this limitation to a large extent. It is unlikely that the external perturbation coincide more than once with the change of set-point.

Another method to prioritize the maintenance actions was presented by Farenzena and Trierweiler (2009) and its main advantage is that there is no need to introduce set-point changes in control loops. Only routine data is required and no further information about the loop is required. This methodology is based on the information on the connection among control loops, which is computed using partial correlation. After obtaining the matrix of partial correlations of data, this method uses the *PageRank* algorithm (Bryan and Leise, 2006).

This work also considered a step in the procedure to contemplate economic and operational aspects to change the priority of the control loops.

Another way to analyse the interaction between control loops is through the **Variability Matrix, VM** (Farenzena et al., 2009). Although it does not require set-point changes, it is necessary to alter the controller. The method produces a matrix that shows the variances of the output variables as a result of controller changes in each loop. This matrix depends on the dynamic behavior of the process, of the controller, and of the external disturbances. The resulting metric shows how the variability is transferred between the control loops and what

impact on specific loop has on the variances of all other loops. This way, the potential economic benefit of each loop can be quantified based on VM.

The works of Rahman and Choudhury (2010) and Rahman and Choudhury (2011) presented another method with two different procedures based on the canonical correlation analysis:

- In the method presented in Rahman and Choudhury (2010), the collected data consists on the controlled variable after set-point steps were applied sequentially in all loops. Then a canonical correlation matrix, M_{cc}, is computed that allows the determination of the order of importance of the loops by using the *PageRank* algorithm (Bryan and Leise, 2006).

- Later, these researchers published a somewhat simplified version of this method (Rahman and Choudhury, 2011). Rahman and Choudhury (2011) calculate the value of the canonical correlation (CCV) between the controlled variable of the control loop where the change of the set-point is made and the group of the output variables of the other control loops. By making a change of set-point in all control loops, a column vector (**canonical correlation vector,** V_{cc}) is obtained. The values of this vector indicate the weight of the relationship with the other control loops. The higher the value of CCV, the higher the interaction is.

The methods that use the canonical correlation are simple but also require at least one change of set-point in all control loops.

Apart from the two procedures referred to above, Rahman and Choudhury (2010, 2011) also suggest a method based on the error (Integral of absolute error, IAE or Integral of squared error, ISE)

Chapter 3

MIMO Control Performance Assessment

In this chapter, a method for MIMO Control Performance Assessment (CPA) is presented. The selection of the method was based on the following points:

- it should disturb as less as possible the process;
- it should be possible to apply such method on-line;
- it should not require the explicit knowledge of dynamic models of the process.
- it would be preferable if the method allowed its generalization to the case of the other benchmark.

This Chapter begins with the analysis of the determination of the interactor matrix. It follows the determination of the system performance including a way to determine a performance range without using the interactor matrix.

3.1 Determination of the interactor matrix

The determination of the interactor matrix is the most complicated step towards the determination of the minimum variance of a MIMO system (Huang, 1997). The procedure to determine this matrix depends on the information available. If the transfer functions of the process are known, the interactor matrix, D, must be calculated using the algorithm developed by Rogozinski et al. (1987). If the process models are not explicitly known, the Markov parameters can be used as initial conditions of Rogozinski's algorithm.

3.1.1 Rogozinski's algorithm

This algorithm is fast, does not present convergence problems and does not depend on the quality of the data. Rogozinski's algorithm is detailed in Appendix B. A summary of this algorithm is set out below.

Consider a matrix of transfer functions G_p with dimension $p \times m$ describing a linear time-invariant plant.

The transfer matrix $G_p(z)$, in domain Z, can be factored according to RMFD (2.6). In this factorization, its numerator is assumed to be a polynomial matrix of the form

$$N(z) = N_0 z^{\theta} + N_1 z^{\theta-1} + \cdots + N_{\theta}. \tag{3.1}$$

The block matrix of the polynomial matrix coefficients is given by

$$N = \begin{bmatrix} N_0 \\ N_1 \\ \vdots \\ N_{\theta} \end{bmatrix}, \tag{3.2}$$

where θ is the degree of the denominator polynomial in RMFD. θ is the order of the interactor matrix. In general, N_i is not a full-rank matrix and, in particular, the leading matrix coefficients $N_0, N_1, \cdots, N_{\theta}$, can be zero. The main steps of the algorithm are as follows:

1. Procedure: Remove all negative exponents of the transfer function matrix.

2. Set $i = 0$ (number of iteration), $N^{(0)}(z) = N(z)$, and $D^{(0)}(z) = I_m$, where I_m is the $m \times m$ identity matrix.

3. If $r_i = \text{rank}(N_0^{i-1}) = \min(r,m)$, terminate and the unitary interactor matrix is $D(z) = D^{i-1}(z)$. N_0 is the polynomial matrix coefficients of the N (RMFD).

   If $r_i < \min(r,m)$, factorize N_0^{i-1} using the QR factorization. The result of this factorization must be

   $$N_0^{(i-1)} = (Q^{(i)})^{-1} \begin{bmatrix} 0_{k_i} \\ N_{0D}^{(i)} \end{bmatrix}, \tag{3.3}$$

   which is equivalent to

   $$Q^{(i)} N_0^{(i-1)} = \begin{bmatrix} 0_{k_i} \\ N_{0D}^{(i)} \end{bmatrix}, \tag{3.4}$$

with $Q^{(i)}$ representing a m $\times$ m non-singular matrix and k_i representing the number of zero lines at the top of the R matrix ($[0_{k_i} N_{0D}^{(i)}]^T$) of the QR factorization ($k_i = p - r_i$). The value of k_i is needed to determine matrix U(z) (see Appendix B).

4. Multiply $N^{(i-1)}(z)$ by matrix Q^i, according to

$$\bar{N}(z) = Q^{(i)} N^{(i-1)}(z), \tag{3.5}$$

in order to obtain the new matrix $\bar{N}(z)$.

5. Multiply $\bar{N}(z)$ by the row shift polynomial matrix, U(z), of order k_i, that is

$$N^{(i)}(z) = U^{(i)}(z)\bar{N}(z). \tag{3.6}$$

This multiplication shifts the matrix of coefficients of $N(z)$ upwards by k_i rows of zeros.

6. Multiply the row shift polynomial matrix (B.2), U(z) by $Q^{(i)}$, that is,

$$S^{(i)}(z) = U^{(i)}(z)Q^{(i)}. \tag{3.7}$$

7. Update the interactor matrix D according to

$$D^{(i)}(z) = S^{(i)}(z)D^{(i-1)}(z). \tag{3.8}$$

Return to step 3 after having determined N_0 to know the rank of this matrix.

When the algorithm stops, matrix D is the unitary interactor matrix and N_0 matrix contains the coefficients of the polynomials in the new N matrix (3.6) which are associated with the exponent equal to the order of the interactor matrix.

This algorithm was implemented in `Octave` (see the code in the Appendix C).

The same example used by Huang (1997) is used here to confirm the correcteness of the code. Consider the 2×2 multivariate process defined by

$$G_p(z^{-1}) = \begin{bmatrix} \frac{z^{-1}}{1-0.4z^{-1}} & \frac{0.5z^{-2}}{1-0.1z^{-1}} \\ \frac{0.3z^{-1}}{1-0.1z^{-1}} & \frac{z^{-2}}{1-0.8z^{-1}} \end{bmatrix}. \tag{3.9}$$

The interactor matrix obtained using the developed program is

$$\mathrm{D}(z) = \begin{bmatrix} -0.9578z & -0.2873z \\ -0.2873z^2 & 0.9578z^2 \end{bmatrix}, \tag{3.10}$$

which coincides with the matrix obtained by Huang (1997), thus suggesting the correctness of the implemented code. In order to study the influence of the process gains on the interactor matrix, several tests were performed. The process gains of the system described by (3.9) were sequentially modified and the corresponding interactor matrices calculated. These process gains are: 1, 0.5, 0.3 and 1. Rewritting the system as

$$\mathrm{G_p}(z^{-1}) = \begin{bmatrix} \frac{\mathrm{K}_{11}z^{-1}}{1-0.4z^{-1}} & \frac{\mathrm{K}_{12}z^{-2}}{1-0.1z^{-1}} \\ \frac{\mathrm{K}_{21}z^{-1}}{1-0.1z^{-1}} & \frac{\mathrm{K}_{22}z^{-2}}{1-0.8z^{-1}} \end{bmatrix}, \tag{3.11}$$

the results obtained were as follows:

- The interactor matrix (3.10) remains the same by changing K_{12} or K_{22} within the range $[-10, 10]$;
- The interactor matrix (3.10) has changed for different K_{11} or K_{21}. The changes observed are indicated in Table 3.1.

Table 3.1: Interactor matrices obtained for several values of K_{11} or K_{21}.

k_{11}	Interactor matrix	k_{21}	Interactor matrix
0	$\begin{bmatrix} 0 & -z \\ -z^2 & 0 \end{bmatrix}$	0	$\begin{bmatrix} z & 0 \\ 0 & z^2 \end{bmatrix}$
0.5	$\begin{bmatrix} -0.8575z & -0.5145z \\ -0.5145z^2 & 0.8575z^2 \end{bmatrix}$	-0.3	$\begin{bmatrix} -0.9578z & 0.2873z \\ 0.2873z^2 & 0.9578z^2 \end{bmatrix}$
2	$\begin{bmatrix} -0.9889z & -0.1483z \\ -0.1483z^2 & 0.9889z^2 \end{bmatrix}$	1	$\begin{bmatrix} -0.7071z & -0.7071z \\ -0.7071z^2 & 0.7071z^2 \end{bmatrix}$
5	$\begin{bmatrix} -0.9982z & -0.0599z \\ -0.0599z^2 & 0.9982z^2 \end{bmatrix}$	5	$\begin{bmatrix} -0.1961z & -0.9805z \\ -0.9805z^2 & 0.1961z^2 \end{bmatrix}$
10	$\begin{bmatrix} -0.9996z & -0.0300z \\ -0.0300z^2 & 0.9996z^2 \end{bmatrix}$	7	$\begin{bmatrix} -0.1414z & -0.9899z \\ -0.9899z^2 & 0.1414z^2 \end{bmatrix}$

The interactor matrix remains unchanged only when the gains changed were the ones of the second column of matrix $\mathrm{G_p}(z^{-1})$ which correspond to the transfer functions presenting bigger delay (z^2). In opposition, changing the process gains of the first column of matrix $\mathrm{G_p}(z^{-1})$

will provoke changes in the interactor matrix. These results show that the gains associated with lower delay cause changes in the delays structure of the process. By increasing the process gain of $G_{p,11}(z^{-1})$, i.e., K_{11}, the values of the main diagonal of the interactor matrix increase while the values of the secondary diagonal decrease. However, its structure was maintained. Increasing K_{21} provokes a decrease of the main diagonal of the interactor matrix while its structure remains the same.

3.1.2 Using Markov parameters

The interactor matrix can be obtained without knowing explicitly the dynamic process models. This method uses the Markov parameters (that is, impulse response coefficients) as initial conditions for the Rogozinski's algorithm. These impulse response coefficients may be obtained directly from industrial data or through time-series models of the form

$$G_p = \frac{1}{1 - a q^{-1}} \quad \text{with} \qquad \frac{1}{1 - a q^{-1}} = \sum_{i=0}^{\infty} a^i q^{-i}. \tag{3.12}$$

To obtain the interactor matrix using only routine data of the process is the biggest advantage of this method. However, this option has also some disadvantages, one of which is the impossibility that sometimes occurs of obtaining the interactor matrix due to computation problems or because the data is not rich enough to extract the information about the process. These problems result from algebraic manipulation of matrices with values close to zero. The other disadvantage is the need to introduce extra noise (**Dither Signal**) during the process stimulation and thus be able to properly calculate the Markov parameters of process.

The first step is to get a time series model that describes the process in open-loop mode. Then, one determines its response to an impulse obtained in order to obtain the Markov model parameters.

The model of the open-loop process is required for this method. However, at industrial level, the data more easily obtained is data in a closed loop situation. Although the Markov parameters of the open-loop and closed-loop transfer function matrix are different, their linear combination yields the same interactor matrix (Huang, 1997). Therefore, the interactor matrix of an open-loop transfer function can be estimated from the closed-loop data.

A brief description of the method is given below. The Markov parameter representation of

a process transfer function matrix can be presented as

$$G_p = \sum_{i=0}^{\infty} G_i q^{-i-1} \tag{3.13}$$

and the interactor matrix as

$$D = D_0 q^{\theta} + D_1 q^{\theta-1} + \cdots + D_{\theta-1} q^{1}, \tag{3.14}$$

where θ is the order of the interactor matrix.

The delay-free transfer function can be represented in the form

$$[D_{\theta-1}, \cdots, D_0] \times \begin{bmatrix} G_0 & 0 & 0 & \cdots & 0 \\ G_1 & G_0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ G_{\theta-2} & G_{\theta-3} & \cdots & \ddots & 0 \\ G_{\theta-1} & G_{\theta-2} & \cdots & \cdots & G_0 \end{bmatrix} = [K, 0, \cdots, 0], \tag{3.15}$$

that is,

$$D^{\circ} G^{\circ} = K^{\circ}. \tag{3.16}$$

G° is termed as the *block-Toeplitz matrix*. If the inequation,

$$\text{rank}(G^{\circ}) \geq \text{rank}(K) = \min(r, m) \tag{3.17}$$

is not satisfied, the block-Toeplitz matrix G° must be expanded by adding more Markov parameters.

Apart from the need to verify (3.17) there is still another condition that must be verified and that will be discussed now (3.21). For determining the order of the interactor matrix θ, which also is the "size" of G°, the *singular value decomposition, (SVD)* technique is used (Huang, 1997; Jelali, 2010), obtaining

$$G^{\circ} = U \Sigma V^{T} = [s_1 s_2] \begin{bmatrix} v_n & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} d_1^T \\ d_2^T \end{bmatrix}. \tag{3.18}$$

The columns of s_2 are the columns of U that are associated with singular values equal to zero in the matrix Σ ($s_2^T G^{\circ} = 0$). v_n is a full rank diagonal matrix with singular values of G°. The rows of d_1^T are the rows of V^T that are associated with singular values not equal to zero in

the matrix Σ. The order of the interactor should be chosen so that d_2^T matrix fulfills the condition

$$K^{\circ} d_2 = 0. \tag{3.19}$$

This equation can be simplified by considering that $K^{\circ} = [K, 0, \cdots, 0]$, such that

$$K^{\circ} d_2 = [K, 0, \cdots, 0] \begin{bmatrix} d_{21} \\ d_{22} \\ \vdots \\ d_{2\theta} \end{bmatrix} = K d_{21}, \tag{3.20}$$

where d_{21} is the upper partition of the matrix d_2 and the number of rows of d_{21} is equal to the number of rows of G_p. It can be concluded that (3.19) is equivalent to

$$K d_{21} = 0. \tag{3.21}$$

If G_p is a square matrix or if it is an $r \times m$ non-square matrix with $r > m$, (3.21) can be simplified to

$$d_{21} = 0. \tag{3.22}$$

If these two conditions (3.21) and (3.22) are not satisfied, the block-Toeplitz matrix must be expanded by adding more Markov parameters until they are satisfied. When these conditions are satisfied, the order of the interactor matrix θ is defined. A block matrix of the first θ Markov parameters is expressed in a block matrix form as

$$\Lambda = [G_0^T, G_1^T, \cdots, G_\theta^T]^T. \tag{3.23}$$

Once this matrix is defined, one applies finally the algorithm proposed by Rogozinski et al. (1987) to calculate the interactor matrix. The numerator matrix coefficients of the RMFD of G_p would be replaced by the first θ Markov parameter matrices.

The methods using the Markov parameters were implemented in `Octave`. The corresponding codes can be found in Appendices F and G where the Markov parameters are obtained from time series models and directly from raw data, respectively. The results obtained determining the Markov parameters from time series models coincide with those obtained in Section 3.1.1, i.e., when knowing explicitly the transfer functions of the process.

The method that computes the interactor matrix directly from the raw data has the advantage

of not requiring the development of a model of the process, but it also has two major drawbacks: there is need to disturb the process with a dither signal and the method presents problems of convergence and requires algebraic manipulation.

To summarize, the dither signal is used to stimulate the process in order to obtain data rich enough to determine the dynamic process models. Once these models are known, the Markov parameters can be found through their impulse responses. The higher the dither signal is, the better the results are. However, a higher dither signal is an undesirable disturbance in the process and thus there is need to reach a compromise when choosing the intensity of the dither signal.

To test this program, data was generated for the structure of the control system shown in Figure 3.1.

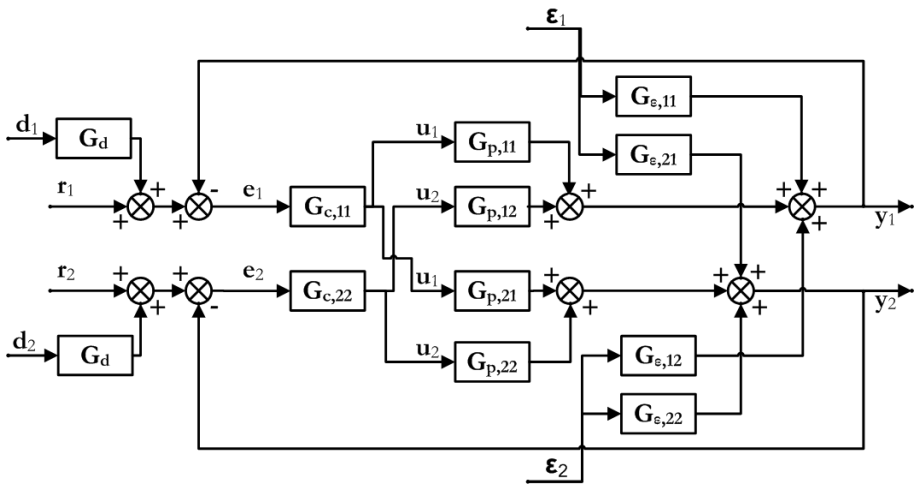


Figure 3.1: Block diagram of a 2×2 system under feedback control with dither signal (d).

The corresponding system transfer functions are

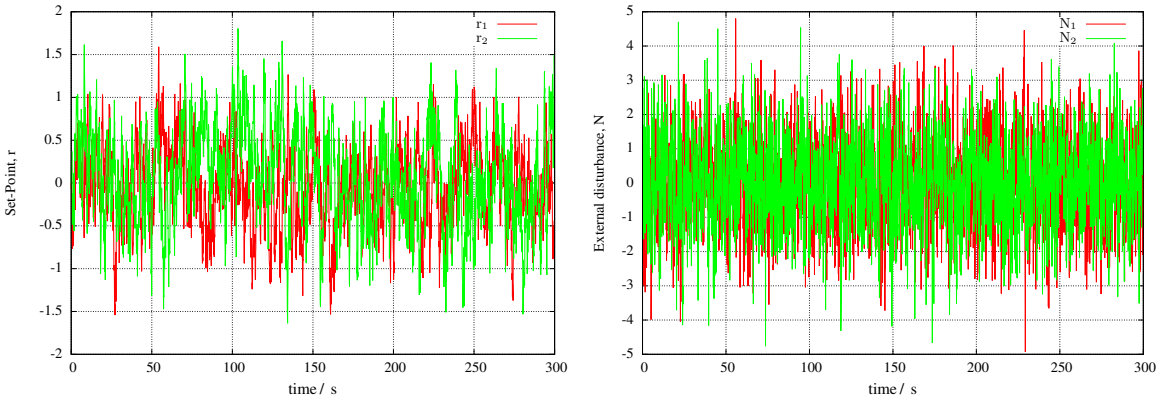
$$\mathrm{G_p}(z^{-1}) = \begin{bmatrix} \frac{z^{-1}}{1-0.4z^{-1}} & \frac{0.5z^{-2}}{1-0.1z^{-1}} \\ \frac{0.3z^{-1}}{1-0.1z^{-1}} & \frac{z^{-2}}{1-0.8z^{-1}} \end{bmatrix}, \qquad \mathrm{G_\varepsilon}(z^{-1}) = \begin{bmatrix} \frac{1}{1-0.5z^{-1}} & \frac{-0.6}{1-0.5z^{-1}} \\ \frac{0.5}{1-0.5z^{-1}} & \frac{1}{1-0.5z^{-1}} \end{bmatrix}. \tag{3.24}$$

In this example, there are two control loops containing two controlled variables, two manipulated variables and two external noise sources. Each controller defines the value of the corresponding manipulated variable based on the error of that control loop. However, the process presents significant interactions between the control loops.

The transfer functions of the controllers are

$$\mathrm{G_c}(z^{-1}) = \begin{bmatrix} \frac{0.5-0.2z^{-1}}{1-0.5z^{-1}} & 0 \\ 0 & \frac{0.25-0.2z^{-1}}{(1-0.5z^{-1})(1+0.5z^{-1})} \end{bmatrix}. \tag{3.25}$$

To test this program, simulated data with 3000 points were used and the sampling time was



(a) Set-points with the dither signals.

(b) External disturbances.

Figure 3.2: Inputs that affect the system.

$\Delta t = 1$s.

For each output variable, the influence of white noise with a variance equal to one ($\sigma_\varepsilon^2 = 1$) is considered. This noise passes through the transfer function G_ε before being applied on variable set-point, r, as it is schematized in Figure 3.1.

In this case, a dither signal is added to the set-point, r. This signal is a white noise sequence that passes through a discrete version of a high-pass filter in the Laplace domain [hfilter$(s) = s/(s+1)$], that is, in the z^{-1} domain

$$G_d(z^{-1}) = \frac{-e^{-\Delta t} z^{-1}}{1 - e^{-\Delta t} z^{-1}}. \tag{3.26}$$

Before passing through the high-pass filter, these two signals (there are two controlled variables) have a variance equal to 0.05 and 0.07, respectively.

Figure 3.2a presents the set-point already including the dither signal. Figure 3.2b shows the process external disturbances, that is, the white noise ε after having passed through the transfer function G_ε while the corresponding evolution of the controlled variable can be observed in Figure 3.3.

This closed loop data was used to identify process models that capture the closed loop dynamics of the system. It is worth mentioning that the results can be affected by the number of parameters chosen for the model. A too high or too low number may cause the prototype to not provide a valid solution. From the experience with the program, the number of parameters should vary between 5 and 15. The data is fitted to a model of type ARX with the number of parameters, mnarx, chosen to be 10. The Appendix D contains the identified model considering just five parameters. It is now possible to determine the Markov parameters. They correspond to the values of the impulse response of the models. Figure 3.4 shows the impulse responses

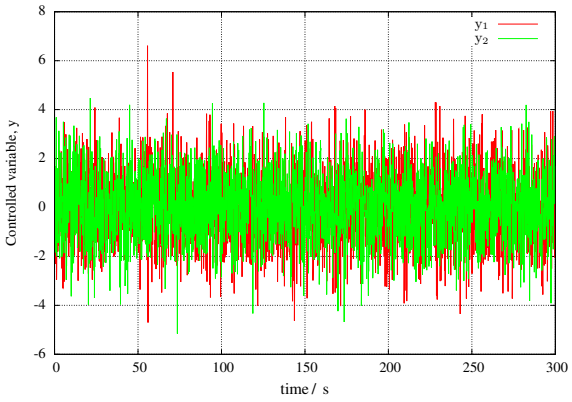


Figure 3.3: Controlled variable under the effect of the dither signal in closed-loop.

(IR) of the model.

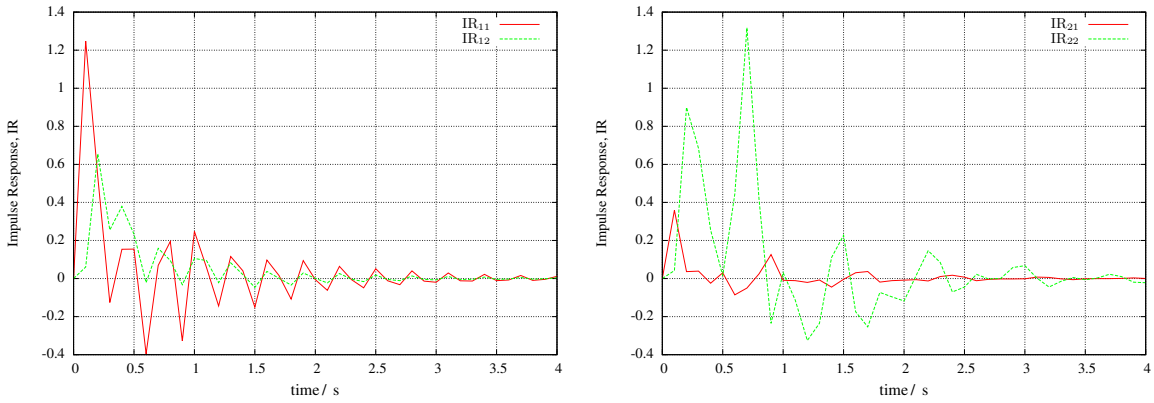


Figure 3.4: Impulse responses of variables y_1 and y_2 relative to u.

It is possible to rewrite the system in the Markov parameter representation, using (3.13). If the matrices containing the Markov parameters have singular values close to zero, a reconditioning method may be applied in order to mitigate potential computational problems. It is an usual procedure "to clean" these matrices before continuing the method. The simplest approach consists of performing SVD of these matrices, and of replacing the values under a certain predefined threshold, a, by zero. The inverse process of the SVD is then applied to obtain the new "clean" matrix with the parameters of Markov. To choose the referred threshold, a rule of thumb is to use either $a = 2\sqrt{N_a}$ or $a = 0.16$, where N_a is the data length (Jelali, 2010). In the present work, both thresholds originated the same results. The interactor matrix obtained is

$$D(z) = \begin{bmatrix} -0.9868z & -0.1617z \\ -0.1617z^2 & 0.9868z^2 \end{bmatrix}. \tag{3.27}$$

Comparing the obtained matrix with the theoretical one (3.10), it is possible to see that their structure is similar with small differences of the coefficients. To improve the estimation of this

matrix, one may increase the dither signal, as it can be seen in Table 3.2, where σ^2_{dither} is the variance of the dither signal (white noise) before passing through the high-pass filter.

Table 3.2: Interactor matrices obtained for various dither signals.

σ^2_{dither}	Interactor matrix
0.005	$\begin{bmatrix} -0.5443z & -0.8389z \\ -0.8389z^2 & 0.5443z^2 \end{bmatrix}$
0.2	$\begin{bmatrix} -0.9554z & -0.2954z \\ -0.2954z^2 & 0.9554z^2 \end{bmatrix}$
0.5	$\begin{bmatrix} -0.9587z & -0.2845z \\ -0.2845z^2 & 0.9587z^2 \end{bmatrix}$

The matrix obtained has a similar structure of that of the theoretical matrix (see matrix 3.10). Although there are slight differences between them, the obtained matrix can be used successfully to determine an approximation of the control system performance.

3.2 Performance assessment method

A method based on routine data for assessing the performance of MIMO control systems was implemented. The developed program requires the following information:

- a unitary interactor matrix determined for the system;
- uncompressed data obtained in a period without set-point changes, normalized to be zero mean.

A brief description of the method implemented is given below. The method comprises the following steps:

1. Identify a multivariate time series model of the closed-loop system, $\mathrm{G_{p,cl}}$.
2. Determine the coefficients of the impulse response (Markov parameters) of the models created in the previous step. Assuming that D is the interactor matrix and θ is its order, the multiplication between $q^{-\theta}\mathrm{D}$ and the process closed-loop model, $q^{-\theta}\mathrm{D} \times \mathrm{G_{p,cl}}$, is equal to a series expansion in which the coefficients are the Markov parameters (F) (3.28).

$$q^{-\theta}\mathrm{D}\,\mathrm{G_{p,cl}} = \mathrm{F}_0 + \mathrm{F}_1 q^{-1} + \cdots + \mathrm{F}_{\theta-1} q^{-\theta+1} \tag{3.28}$$

3. Determine the covariance matrix of the noise Σ_ε via (2.12) or using the residues resulting from the obtained model, $G_{p,cl}$.

4. Calculate the minimum variance covariance matrix for the interactor-filtered output $\tilde{Y}$ as

$$\tilde{\Sigma}_{mv} = F_0\Sigma_\varepsilon F_0^T + F_1\Sigma_\varepsilon F_1^T + \cdots + F_{\theta-1}\Sigma_\varepsilon F_{\theta-1}^T, \tag{3.29}$$

   where $\tilde{Y} = q^{-\theta}D(q)Y$.

5. Estimate the actual variance of the process, Σ_Y.

6. Since D is a unitary interactor matrix, the minimum variance control law that minimizes the objective function of the interactor-filtered variable, $\tilde{Y}$, also minimizes the objective function of the original variable, Y. Thus, the multivariate performance index can be calculated as

$$\eta_{cl} = \frac{\text{trace}(\tilde{\Sigma}_{mv})}{\text{trace}(\Sigma_Y)}. \tag{3.30}$$

This is the index of overall performance. To calculate the performance indices for each output variable it is necessary to perform the following steps:

1. Write the unitary interactor matrix in the form

$$D = D_0 q^\theta + D_1 q^{\theta-1} + \cdots + D_{\theta-1}q. \tag{3.31}$$

2. Calculate the matrix $E = [E_0, E_1, \cdots, E_{\theta-1}]$ as

$$[E_0, E_1, \cdots, E_{\theta-1}] = [D_0^T, D_1^T, \cdots, E_{\theta-1}^T] \times \begin{bmatrix} F_0 & F_1 & \cdots & F_{\theta-2} & F_{\theta-1} \\ F_1 & F_2 & \cdots & F_{\theta-1} & 0 \\ \vdots & \vdots & \ddots & \vdots & 0 \\ \vdots & F_{\theta-1} & \cdots & \vdots & \\ F_{\theta-1} & 0 & \cdots & 0 & 0 \end{bmatrix}. \tag{3.32}$$

3. Calculate the minimum variance covariance matrix of the original output Y by

$$\Sigma_{mv} = E_0\Sigma_\varepsilon E_0^T + E_1\Sigma_\varepsilon E_1^T + \cdots + E_{\theta-1}\Sigma_\varepsilon E_{\theta-1}^T. \tag{3.33}$$

The performance indices for each individual output can be calculated as

$$[\eta_{y1}, \eta_{y2}, \cdots, \eta_{yp}] = [\text{diag}(\Sigma_{mv})] \cdot \times [\text{diag}(\Sigma_Y)]^{-1}, \tag{3.34}$$

where $\cdot\times$ is an element-by-element multiplication operator. The second term of this multiplication is the inverse of the real covariance matrix with off-diagonal values set to zero.

This algorithm is the basis of the FCOR method used to multivariable systems (MIMO FCOR) and was implemented in `Octave`. In the Appendix H, a detailed description of this algorithm is presented and the code may be found in Appendix I.

Results

To test this prototype, two sets of simulated data (A and B) were used. The datasets A and B represent the same process G_p (3.9) and the same controller G_c (3.25). The only difference between them is the matrix of transfer functions from external disturbances G_ε:

$$G_{\varepsilon A}(z^{-1}) = \begin{bmatrix} \frac{1}{1-0.5z^{-1}} & \frac{-z^{-1}}{1-0.6z^{-1}} \\ \frac{z^{-1}}{1-0.7z^{-1}} & \frac{1}{1-0.8z^{-1}} \end{bmatrix}, \qquad G_{\varepsilon B}(z^{-1}) = \begin{bmatrix} \frac{1}{1-0.5z^{-1}} & \frac{-0.6}{1-0.5z^{-1}} \\ \frac{0.5}{1-0.5z^{-1}} & \frac{1}{1-0.5z^{-1}} \end{bmatrix}. \tag{3.35}$$

For each of the matrices (3.35), three simulations were performed to verify the robustness of the method. The obtained simulated data were processed using the prototype implementation shown in Appendix I and the results are given in Tables 3.3 and 3.4.

The dataset consists of 3000 points with the sampling time (Δt) of 0.1s. For each output variable, the external disturbance is white noise with a variance equal to one $(\sigma_\varepsilon^2 = 1)$. The dither signal before passing through the high-pass filter is white noise with a variance of 0.2 (see table 3.2).

Table 3.3: Performance metrics using data with $G_{\varepsilon A}$.

η_Y	η_{y_1}	η_{y_2}	Cov(Y)	Cov(Y_{mv})
0.54	0.41	0.65	$\begin{bmatrix} 3.1997 & -0.6917 \\ -0.6917 & 3.9237 \end{bmatrix}$	$\begin{bmatrix} 1.3055 & -0.4478 \\ -0.4478 & 2.5452 \end{bmatrix}$
0.57	0.45	0.67	$\begin{bmatrix} 3.1498 & -0.7787 \\ -0.7787 & 3.7758 \end{bmatrix}$	$\begin{bmatrix} 1.4109 & -0.6451 \\ -0.6451 & 2.5317 \end{bmatrix}$
0.56	0.46	0.64	$\begin{bmatrix} 3.1858 & -0.7514 \\ -0.7514 & 3.9954 \end{bmatrix}$	$\begin{bmatrix} 1.4670 & -0.6261 \\ -0.6261 & 2.5540 \end{bmatrix}$

In Tables 3.3 and 3.4, η_Y is the value of the overall performance of the MIMO system, η_{y_1}

and η_{y_2} are the values of the performance of the variables y_1 and y_2, Cov(Y) is the covariance matrix of the output variables, and Cov(Y_{mv}) is the covariance matrix of the output variables of the process with a minimum variance controller.

Table 3.4: Performance metrics using data with $G_{\varepsilon B}$.

η_Y	η_{y_1}	η_{y_2}	Cov(Y)	Cov(Y_{mv})
0.72	0.63	0.84	$\begin{bmatrix} 2.3781 & 0.0879 \\ 0.0879 & 1.8357 \end{bmatrix}$	$\begin{bmatrix} 1.5004 & -0.1634 \\ -0.1634 & 1.5522 \end{bmatrix}$
0.74	0.65	0.85	$\begin{bmatrix} 2.3379 & 0.1132 \\ 0.1132 & 1.8887 \end{bmatrix}$	$\begin{bmatrix} 1.5178 & -0.1696 \\ -0.1696 & 1.6029 \end{bmatrix}$
0.74	0.64	0.87	$\begin{bmatrix} 2.3650 & 0.0777 \\ 0.0777 & 1.7087 \end{bmatrix}$	$\begin{bmatrix} 1.5138 & -0.1389 \\ -0.1389 & 1.4918 \end{bmatrix}$

These results indicate that the controller performance is rather different from the desired values (corresponding to the minimum variance situation) in both cases, and that it is higher in case B. Besides, the obtained results are robust due to the fact that there is a good dither signal (chosen according to the results shown in Table 3.2).

The values obtained for the overall performance of the control system in Tables 3.3 and 3.4 consider only the main diagonal values of the covariance matrices, that is, this index considers only the variability of the controlled variables originated by their own loops, neglecting the variability due to the interaction between them. Therefore, this index is suitable when there is little interactions among the control loops. In the case of significant interaction, it is better to apply another index that uses more information of the covariance matrices, for instance, via the determinant of the covariance matrix I_v (2.16). The determinant of this matrix is a measure of the variability of a subspace that includes both controlled variables. Table 3.5 compares the performance metrics obtained using the diagonal elements with those obtained using the determinant. The column of η_Y contains the values of the original method (exposed in detail in Tables 3.3 and 3.4).

Table 3.5: Comparison of the values obtained by the original method with the values obtained by this new index.

G_ε	η_Y	I_v
A	0.54	0.26
A	0.56	0.28
A	0.57	0.28
B	0.72	0.53
B	0.74	0.55
B	0.74	0.55

These values show a greater penalization of the difference of variances with this new index. This new index shows that this controller has a poor performance for the two control systems (significantly less than one). This penalization increases when the interaction between the control loops increase, as can be seen in Table 3.6, where the values of the performance of the control system are compared with two levels of interaction between control loops. These values were obtained for simulated data from the transfer function matrix

$$G_p(z^{-1}) = \begin{bmatrix} \frac{z^{-1}}{1-0.4z^{-1}} & \frac{K_{12}z^{-2}}{1-0.1z^{-1}} \\ \frac{0.3z^{-1}}{1-0.1z^{-1}} & \frac{z^{-2}}{1-0.8z^{-1}} \end{bmatrix}. \tag{3.36}$$

Increasing the value of K_{12} ($K_{12} = 0.5$ in G_p of the equation 3.24), the interaction between the two loops increases.

Table 3.6: Comparison of performance for two levels of interaction between control loops.

K_{12}	η_Y	I_v
0.5	0.541	0.259
0.5	0.560	0.276
7	0.272	0.096
7	0.249	0.082

Table 3.6 clearly shows that the relative difference between the two indices is much higher when the interaction between the control loops is higher. Therefore, this new index reinforces the conclusions already taken with the first index, that the controllers must be retuned or replaced by a more adequate / advanced one.

The influence of loop interaction on the performance index is illustrated in Figure 3.5. It shows that the performance is impacted increasing the level of interaction between the control loops and that most of this impact is due to variable y_1. Although the interaction factor is directly associated with variable y_1, it affects the whole process.

The most difficult part of the described performance assessment methods is the determination of the interactor matrix as this process depends on the quality of the data. As it was shown in Section 3.1.2, it is possible to increase the information content of the data using a dither signal. However, such approach may be infeasible in certain processes.

In these cases, one should use a method that does not require the interactor matrix. This method is based on the calculation of a lower and an upper bounds of multivariable minimum variance. It is possible to derive such bounds based on the knowledge of the order of the interactor matrix and from the time delays between inputs and outputs (Jelali, 2010). The lower bound of the overall performance index is calculated assuming that the interactor matrix

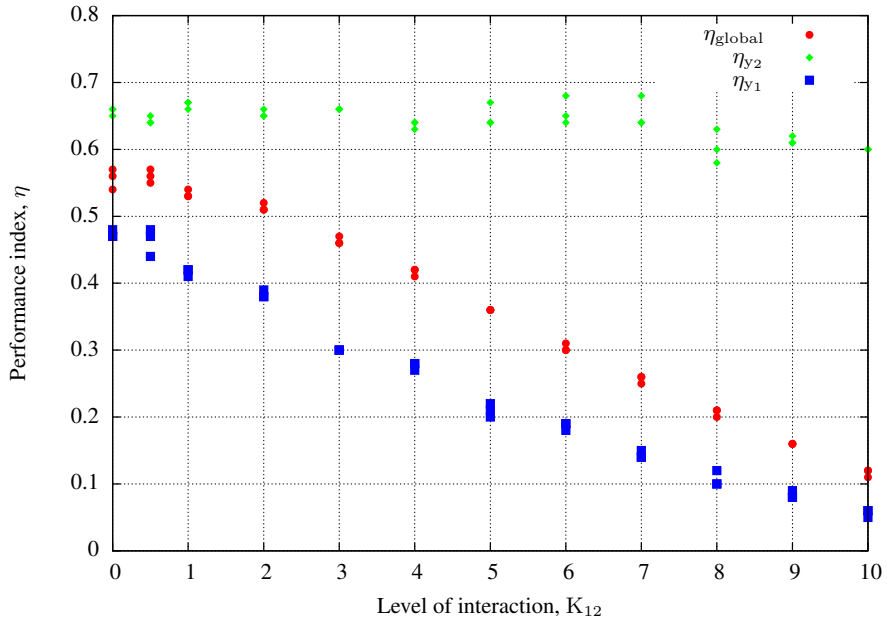


Figure 3.5: The relationship between the performance of the process with the level of interaction.

is a diagonal matrix where each element is the value of the smallest delay for this variable (smallest value of delay in each row of the process transfer functions matrix). The upper bound of overall performance index is calculated assuming that the interactor matrix is the diagonal matrix $\mathrm{D} = z^{\theta}\mathrm{I}$, with θ representing the value of the order of the interactor matrix or the largest time delay in the case the order is not known. It is worth mentioning that to know the order of the interactor matrix it is not compulsory to know the interactor matrix itself. There are methods to determine this order without the explicit knowledge of the interactor matrix (Jelali, 2010).

To apply this method there is only need of the time delay of each output variable with respect to each input and the order of the interactor matrix. These delays are easily obtained using open-loop data. Let one consider the data of the process described earlier in this Section (page 33). The order of the interactor matrix is considered to be $\theta = 2$ (3.10). The lower bound of overall performance index is calculated assuming that the interactor matrix is

$$\mathrm{D} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{3.37}$$

Table 3.7 presents the results obtained for the same two data sets also used to obtain the results presented in Tables 3.3 and 3.4.

The obtained values are consistent with those calculated with the interactor matrix. In this case, the best estimate is the lower bound of the performance. The values obtained show that this method provides a good estimate, the two bounds have values significantly lower when

Table 3.7: Performance of the control system without the interactor matrix.

G_ε	η_{upper}	η_{lower}
A	0.86	0.46
A	0.84	0.45
A	0.85	0.45
B	0.93	0.71
B	0.92	0.71
B	0.92	0.71

using $G_{\varepsilon,A}$, as it happens when using the interactor matrix. The interval of performance values is larger when process performance is lower.

Chapter 4

Interaction in MIMO Systems and Prioritization of Loop Maintenance

Interaction between control loops negatively affects the performance of the overall control system of processes. The problems of a certain control loop can easily expand and deteriorate the performance of controllers from other interacting loops. In this context, it is very important to establish the order according to which the problematic loops should go under maintenance, giving priority to the loops that present stronger interaction. It is, therefore, essencial to analyze and somehow quantify the level of interaction. In this chapter, this subject is tackled in a comparative way through different available methods. With that purpose, different sets of data were generated.

The methods used were:

- IAE modified technique by Rossi et al. (2006)
- *LoopRank* method by Farenzena and Trierweiler (2009)
- Variability matrix by Farenzena et al. (2009)
- Using Canonical correlation by Rahman and Choudhury (2011)
- Using IAE or ISE by Rahman and Choudhury (2011)

4.1 Data generation

Consider the closed loop system depicted in Figure 4.1, where the process has three output variables, three input variables and two external noise sources.

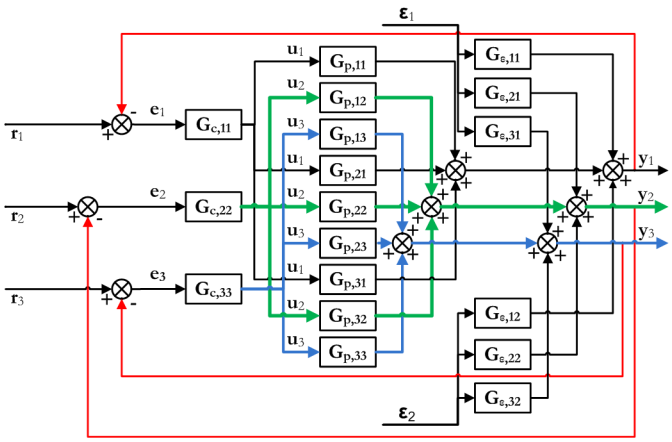


Figure 4.1: Block diagram of the feedback control loops of the system.

Datasets from three different processes ($G_{p,A}$, $G_{p,B}$ and $G_{p,C}$) in a closed-loop as the one represented in Figure 4.1 were generated. The first process, $G_{p,A}$ is the process presented by Rossi et al. (2006), designated by *Shell Problem*. The transfer functions of $G_{p,A}$, continuous and in the Laplace domain, are

$$G_{p,A} = \begin{bmatrix} \frac{4.5e^{-27s}}{50s+1} & \frac{1.77e^{-28s}}{60s+1} & \frac{5.88e^{-27s}}{50s+1} \\ \frac{5.39e^{-18s}}{50s+1} & \frac{5.62e^{-14s}}{60s+1} & \frac{6.9e^{-15s}}{50s+1} \\ \frac{4.38e^{-20s}}{33s+1} & \frac{4.42e^{-22s}}{44s+1} & \frac{7.2}{19s+1} \end{bmatrix}. \tag{4.1}$$

The delays of these transfer functions were approximated in the simulations using the *Padé* approximation

$$e^{-\theta s} \approx \frac{1 - \frac{\theta}{2}s}{1 + \frac{\theta}{2}s} \tag{4.2}$$

For the purposes of the present study, the external disturbances of the process are neglected. For most of the techniques to be analysed here it is necessary to stimulate the closed-loop system with a step in the set-point. But the effect of an external disturbance is, in general, insignificant when compared with the effect provoked by the change in the set-point. The three controllers are PID's that were tuned firstly using the rules of Cohen-Coon (as done by Seborg et al., 2004) and then this tuning was refined by trial and error around the parameters previously found. The obtained PI controllers are

$$G_c = \begin{bmatrix} \frac{25s+0.25}{100s} & 0 & 0 \\ 0 & \frac{65s+0.65}{100s} & 0 \\ 0 & 0 & \frac{15s+1.5}{10s} \end{bmatrix}. \tag{4.3}$$

For data generation, 2500 points were used with a sampling time $\Delta t = 1$s. The responses of

the three output variables to a unitary change in the set-point of each control loop were simulated (a change at a time). This produced three sets of responses with three output variables per set. The evolution of the controlled and of the manipulated variables for these three simulations can be seen in Figures 4.2, 4.3 and 4.4, with stimulation of set-point of control loops 1, 2 and 3, respectively.

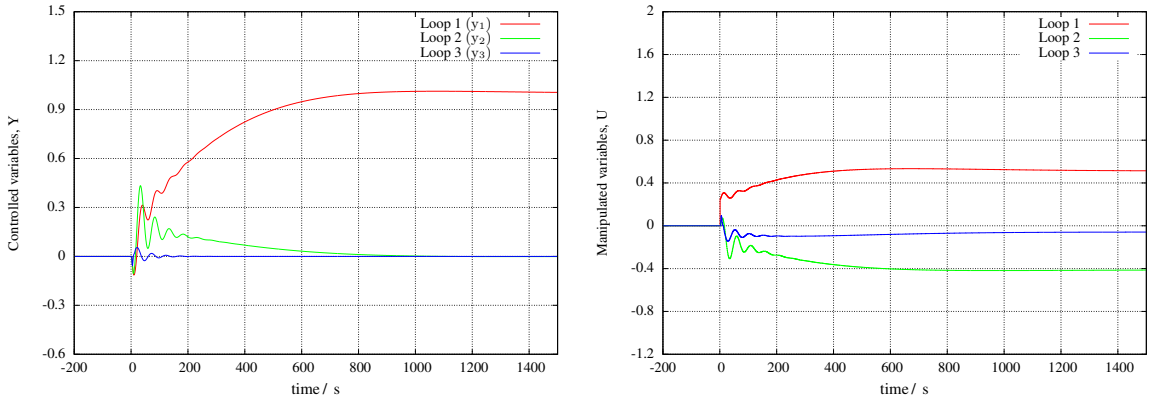


Figure 4.2: Simulation of the controlled and manipulated variables in response to a step change in the set-point of loop 1.

It is possible to see in Figure 4.2 that the controlled variable of loop 1 moves towards its new set-point. But it is also clear that the three control loops have interactions between them and the other two variables moved temporarily away from their set-points as an exclusive consequence of such interaction (their set-points kept unchanged). All the three variables responded in the same direction.

Figure 4.3 shows the system evolution when there is a unit change in the set-point of loop 2.

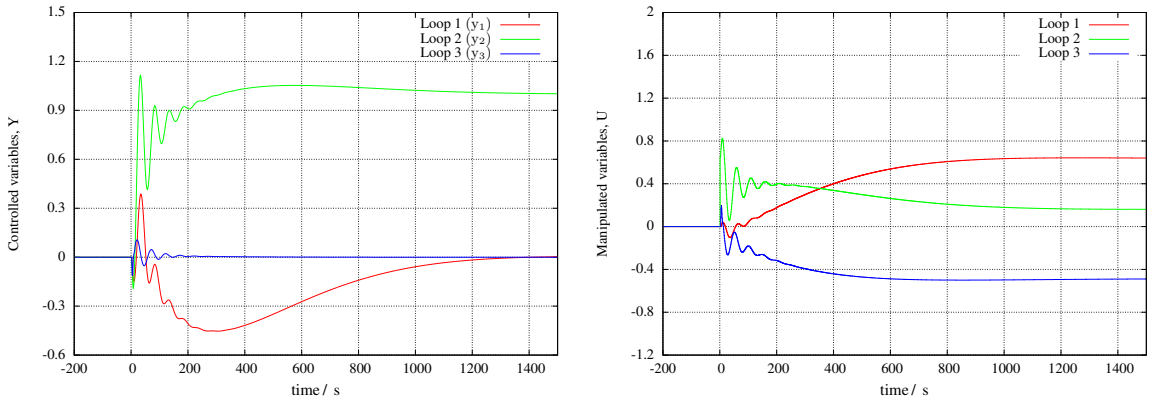


Figure 4.3: Simulation of the controlled and manipulated variables in response to a step change in the set-point of loop 2.

Again, it is possible to identify the existence of strong interactions between loop 2 and the other control loops, especially loop 1. An increase of variable y_2 induces a decrease in

variable y_1 that will be eliminated only later on. There is a stronger interaction between loops 2 and 1 than between loops 2 and 3.

Finally, Figure 4.4 shows the evolution of the closed-loop system under the effect of a change in the set-point of loop 3.

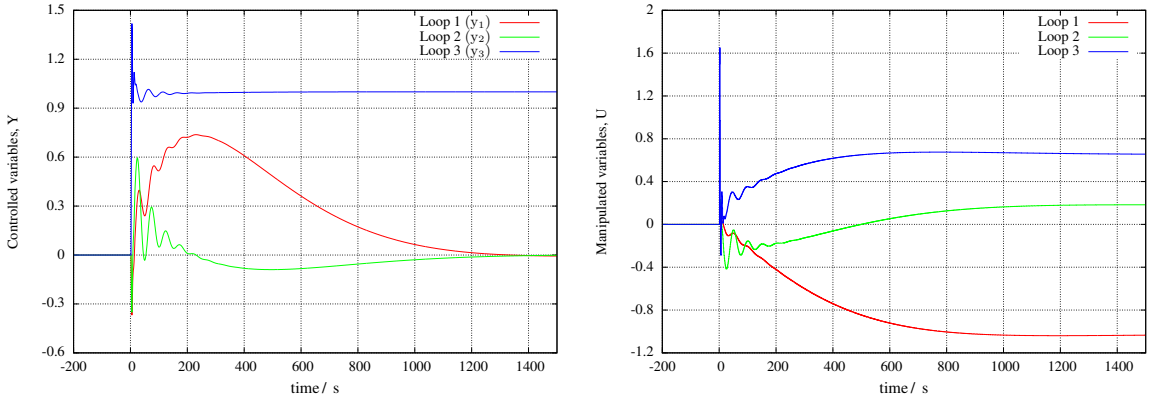


Figure 4.4: Simulation of the controlled and manipulated variables in response to a step change in the set-point of loop 3.

It is also possible to identify the existence of strong interactions between loop 3 and the other control loops, especially loop 1. This interaction with loop 1 is the largest of all the set-point changes. The loop 2 is more affected with this set-point change. The second process used for data generation is $G_{p,B}$. All the transfer functions related to the interactions between control loops are the same except the value of the gain which is bigger for interactions related to loop 1 (gain $= 5$) and lower for interactions related to loop 3 (gain $= 0.5$). This process was designed with the purpose of creating a process to which the level of interaction between loops was obvious. For process $G_{p,B}$ the loop affecting more the others is loop 1, followed by loop 2 and finally by loop 3. Only the transfer functions on the main diagonal (direct relationship between input and output variables of the same loop) have delay, equal to the three control loops. Maintaining the delay in the three transfer functions, eases the analysis, since the results are not distorted by different delays. This process has three output variables and three input variables. The transfer functions between them, continuous and in the Laplace domain, are

$$G_{p,B} = \begin{bmatrix} \frac{4.0e^{-2s}}{20s+1} & \frac{2}{50s+1} & \frac{0.5}{50s+1} \\ \frac{5}{50s+1} & \frac{10.0e^{-2s}}{25s+1} & \frac{0.5}{50s+1} \\ \frac{5}{50s+1} & \frac{2}{50s+1} & \frac{1.5e^{-2s}}{10s+1} \end{bmatrix}. \tag{4.4}$$

The delays of these transfer functions were approximated in the simulations using the *Padé* approximation (4.2). The simulated closed-loop data was obtained with simple PI controllers

whose transfer functions are

$$G_{c,B} = \begin{bmatrix} \frac{25s+1.25}{20s} & 0 & 0 \\ 0 & \frac{15.62s+0.62}{25s} & 0 \\ 0 & 0 & \frac{16s+1.6}{10s} \end{bmatrix}. \tag{4.5}$$

For data generation, the system was stimulated with unitary changes in the set-points of each control loop, one at each time. For each stimulus, 1000 points were used with a sampling time $\Delta t = 1$s and the responses of the three output variables were registered. The generated data can be observed in Figure 4.5.

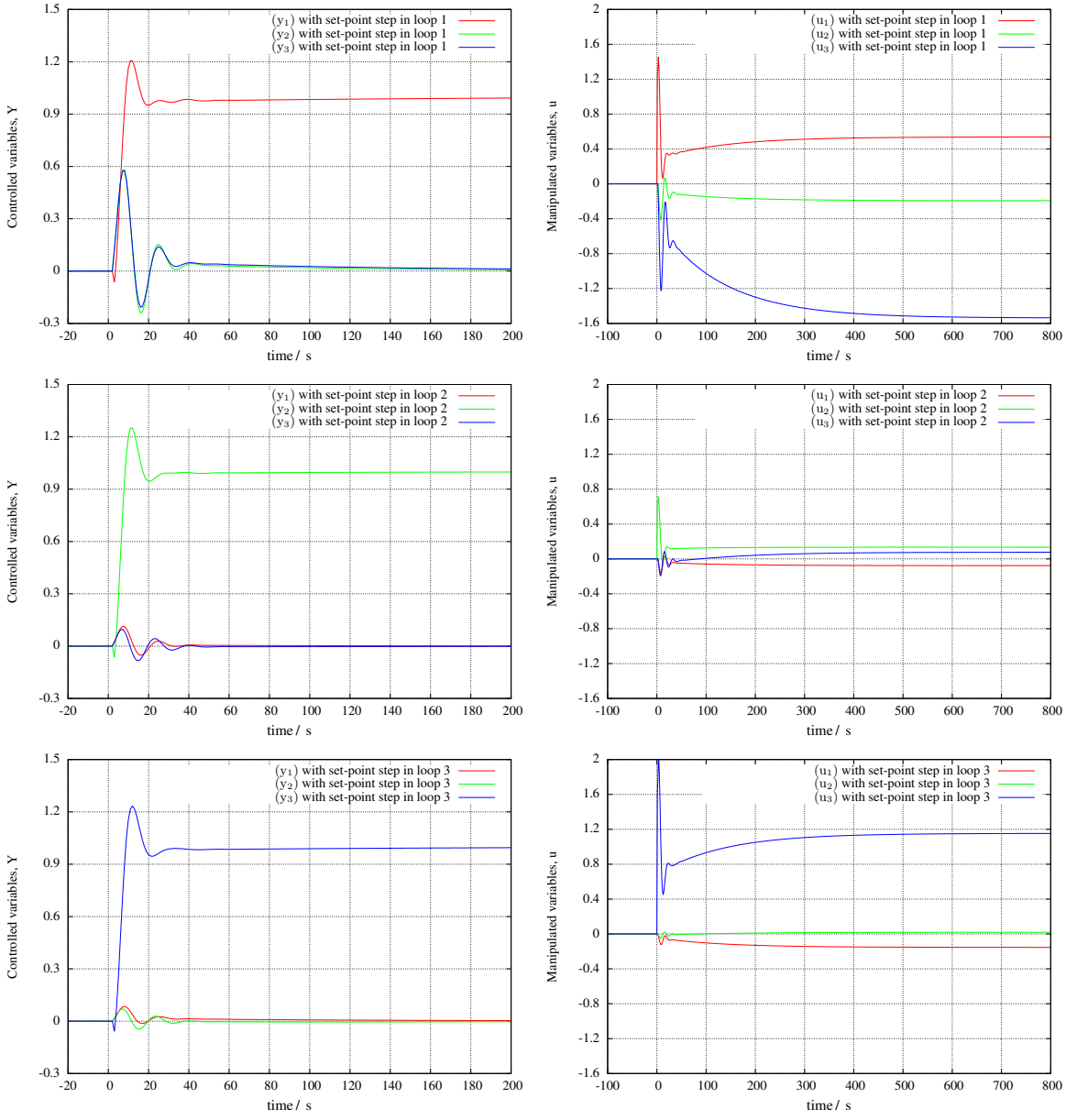


Figure 4.5: $G_{p,B}$ response to sequential unitary changes in the set-points.

Still with process $G_{p,B}$, another dataset was generated by stimulating the closed-loop process

with external noise with unit variance while maintaining all the set-points. The multivariable disturbance model is

$$G_{\varepsilon,B} = \begin{bmatrix} \frac{1.2}{45s+1} & \frac{1.44}{40s+1} \\ \frac{1.52}{25s+1} & \frac{1.83}{20s+1} \\ \frac{1.14}{27s+1} & \frac{1.26}{32s+1} \end{bmatrix}. \tag{4.6}$$

The calculation of the variability matrix requires the comparison of the system response with the response that it would have if the controllers were different. These new controllers used to determine the variability matrix (used not simultaneously) are

$$G_{c,B,new} = \begin{bmatrix} \frac{40s+2}{20s} & 0 & 0 \\ 0 & \frac{25s+1}{25s} & 0 \\ 0 & 0 & \frac{22s+2.2}{10s} \end{bmatrix}. \tag{4.7}$$

The third process from which data was generated, $G_{p,C}$, has transfer functions very similar with

$G_{p,B}$. The only difference is that $G_{p,C}$ does not have any delay. As the dynamics of the process is different (no delays in the process), the controllers needed to be a bit different compared with the controllers used to control $G_{p,B}$. The simulated closed-loop data was obtained using controllers whose transfer functions are

$$G_{c,C} = \begin{bmatrix} \frac{50.0s+2.5}{20s} & 0 & 0 \\ 0 & \frac{31.2s+1.2}{25s} & 0 \\ 0 & 0 & \frac{33.0s+3.3}{10s} \end{bmatrix}. \tag{4.8}$$

For data generation, 1000 points were used with a sampling time $\Delta t = 1$s. The same procedure that was done for the first process, was used. The profiles of the three output and manipulated variables responding to unitary changes in the set-points, can be observed in Figure 4.6.

The multivariable disturbance model obtain the variability matrix is the same (4.6) and the controllers used sequentially for comparison purposes in order to build the variability matrix are

$$G_{c,C,new} = \begin{bmatrix} \frac{60.0s+3}{20s} & 0 & 0 \\ 0 & \frac{43.8s+1.8}{25s} & 0 \\ 0 & 0 & \frac{40.0s+4.0}{10s} \end{bmatrix}. \tag{4.9}$$

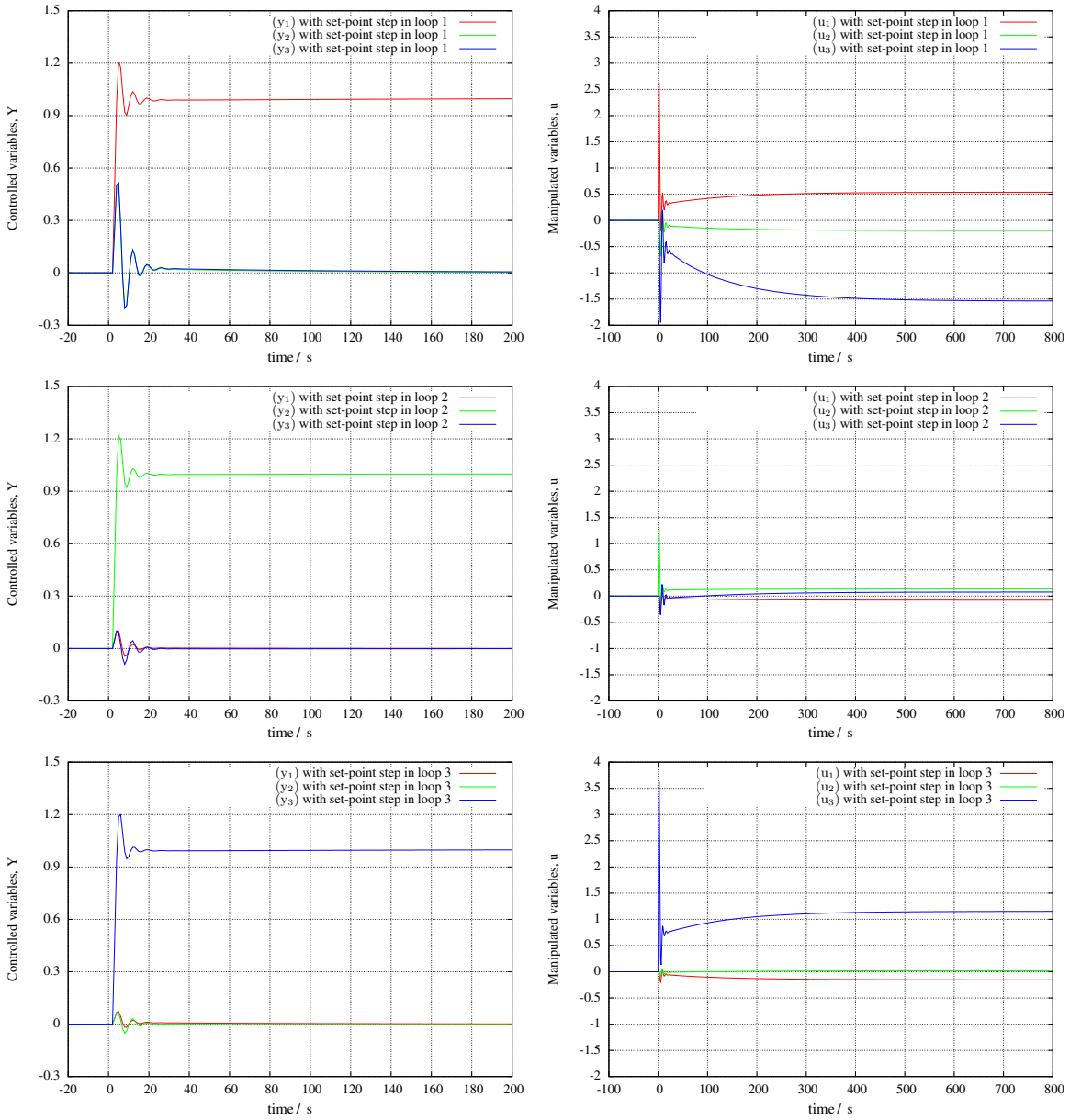


Figure 4.6: Output and controlled variable profiles with a unitary change in the set-point.

4.2 Interaction Analysis and Loop Prioritization

Several methods were used to conduct the interaction analysis. Each of them is described and applied to the first dataset. In the end, the results of the three datasets is presented.

Using IAE or ISE This method was suggested by Rahman and Choudhury (2011) and it is based on an analysis of the error (integral of absolute value of the error, IAE, or integral of squared error, ISE). These errors are defined as

$$\mathrm{IAE} = \int_0^\infty \mid \mathrm{e}(t) \mid \mathrm{d}t \tag{4.10}$$

and

$$\mathrm{ISE} = \int_0^\infty \mathrm{e}^2(t)\mathrm{d}t, \tag{4.11}$$

where $\mathrm{e}(t) = \mathrm{y_{sp}}(t) - \mathrm{y}(t)$ is the error.

A control loop with a significant interaction has a value of IAE or ISE higher when there is a disturbance of the other control loops. The method consists on stimulating the system with set-point changes in the various loops and to compare the error associated with the controlled variable of the loop changed with the error of controlled variables of the other loops.

The loop interaction (LI) between loops i and j is calculated by

$$\mathrm{LI}_{i,j} = \frac{\mathrm{IAE}_j}{\mathrm{IAE}_i}, \tag{4.12}$$

for a situation of having changed the set-point of loop i. Alternatively, the loop interaction can be quantified by the ratio of ISE error through a similar procedure.

The interaction between all control loops can be summarized through the loop interaction matrix (M) containing the individual values computed according to 4.12. That is,

$$\mathrm{M} = \begin{bmatrix} \mathrm{LI}_{1,1} & \cdots & \mathrm{LI}_{1,\mathrm{n_c}} \\ \vdots & \ddots & \vdots \\ \mathrm{LI}_{\mathrm{n_c},1} & \cdots & \mathrm{LI}_{\mathrm{n_c},\mathrm{n_c}} \end{bmatrix}. \tag{4.13}$$

This matrix is square with dimension equal to the number of control loops, $\mathrm{n_c}$.

Each row of matrix M indicates how a certain loop interacts with all the others. Each column represents how the various loops interact with a specific loop. Once having constructed matrix M, it is possible to rank the loops according to their interactions importance. Such classification appears in the form of a vector, called **Loop Importance Index (LII)**.

The loop importance index is calculated from the loop interaction matrix (M) according to

$$\mathrm{LII}_i = \frac{\sum_{j=1}^{\mathrm{n_c}} \mathrm{LI}_{i,j}}{\max\left(\sum_{j=1}^{\mathrm{n_c}} \mathrm{LI}_{k,j}\right)} \times 100\% \quad (\text{with} \quad k = 1, \cdots, \mathrm{n_c}). \tag{4.14}$$

Each value of the loop importance index is the ratio between the sum of each row and the largest sum that is obtained for the various rows of matrix M, ie, corresponds to the sums of the rows of M normalized by the biggest of these values. The values of the loop importance index range from 0 to 100%.

Calculating the interaction matrix for the data of the simulation described in Section 4.1, one obtains

$$M_{IAE} = \begin{bmatrix} 1.00 & 0.30 & 0.01 \\ 3.03 & 1.00 & 0.07 \\ 48.28 & 9.59 & 1.00 \end{bmatrix} \text{ or } \quad M_{ISE} = \begin{bmatrix} 1.00 & 8.24 \times 10^{-02} & 5.76 \times 10^{-04} \\ 2.81 & 1.00 & 8.40 \times 10^{-03} \\ 81.00 & 4.40 & 1.00 \end{bmatrix}. \quad (4.15)$$

The first, second and third columns of both matrices are calculated using the data showed in Figures 4.2, 4.3 and 4.4, respectively. The loop importance index,LII, is

$$LII_{IAE} = \begin{bmatrix} 2.23 \\ 6.96 \\ 100.00 \end{bmatrix} \text{ or } \quad LII_{ISE} = \begin{bmatrix} 1.25 \\ 4.42 \\ 100.00 \end{bmatrix}. \quad (4.16)$$

These results clearly show that the third loop is significantly more important than loops 1 and 2. The loops can be ranked by order of importance as loop 3, loop 2 and loop 1, which will be abbreviated as $3 > 2 > 1$.

A visual analysis of Figures 4.2, 4.3 and 4.4 corroborates this conclusion. These Figures show that the first loop has the lowest interaction, ie, when making a change in the set-point of loop 1, (Figure 4.2), variable y_3 almost does not change and variable y_2 has an initial peak of magnitude lower and takes less time to stabilize than in the case where the change of the set-point occurred in loop 3 (Figure 4.4). Loop 1 is significantly affected by the other two loops, specially by loop 3 (see Figures 4.3 and 4.4). The loop that produces smaller changes in the controlled variables of the other loops is loop 1 (in variables y_2 and y_3), followed by loop 2 (in variables y_1 and y_3) and, finally, by loop 3 (in variables y_1 and y_2). A change in loop 1 provoked IAEs of 0.3 and 0.01 in the other loops. A change in loop 2 provoked IAEs of 3 and 0.07 in the controlled variables of the other loops. A change in loop 3 provokes IAEs of 48 and 10 in the controlled variables of the other loops. Therefore, the loops can be ordered as $3 > 2 > 1$.

Although the same conclusion is achieved using either IAE or ISE, ISE amplifies the error values and, therefore, there is a greater difference between the coefficients in the loop importance index computed through ISE. The order indicated above ($3 > 2 > 1$) corresponds to the most appropriate order to carry out loop maintenance.

Through canonical correlation The canonical correlation method has two different versions, both suggested by Rahman and Choudhury (2011, 2010):

- The first version consists on computing the scalar value of the canonical correlation (CCV) between the controlled variable of the control loop where the change in the set-point is made and the controlled variables of the remaining loops. This calculation is repeated performing the change of set-point consecutively in all loops. The various CCV values are then collected in a vector (**canonical correlation vector,** V_{cc}), with size equal to the number of control loops,

$$V_{cc} = \begin{bmatrix} CCV_1 \\ CCV_2 \\ \vdots \\ CCV_{n_c} \end{bmatrix}. \tag{4.17}$$

The values of this vector give an indication of how strongly the loops are related. The higher the value of CCV_i, the higher is the interaction between loop i and the remaining loops. The last step is to rank the loops according to their importance. This can be performed by normalizing each element of the canonical correlation vector by the maximum element of that vector and expressing it in percentage. A value of 100% indicates the loop with maximum interaction, thus requiring to be the first to be scheduled for maintenance.

Applying this method to the example of Section 4.1, the canonical correlation vector obtained is

$$V_{cc} = \begin{bmatrix} 0.76 \\ 0.10 \\ 0.08 \end{bmatrix}. \tag{4.18}$$

The corresponding **loop importance index,**(LII_{cc}) is

$$LII_{cc} = \begin{bmatrix} 100.00 \\ 13.59 \\ 10.68 \end{bmatrix}. \tag{4.19}$$

The control loops can be ranked as $1 > 2 > 3$. The loop importance index vector clearly shows that the first loop is significantly more important than the others.

Interestingly, this result is in disagreement with the one obtained by the previous method for the same data (see page 47). It also disagrees with the result reported by the authors of the method (Rahman and Choudhury, 2011). The closed-loop systems of the present

work and that of Rahman and Choudhury (2011) are not exactly the same since Rahman and Choudhury (2011) did not mention what controllers had been used in closed-loop and how had they been tuned. Therefore, in the present work a PID was used and tuned. Anyway, the behavior of the closed-loop system of the present work reproduces very well the behavior exhibited by the closed-loop system of Rahman and Choudhury (2011). Therefore, the controllers used here should not be very different of the ones used by Rahman and Choudhury (2011).

In order to explore the referred differences of results, some further analysis and tests were carried out. One of those tests consisted on repeating the calculations using data generated in a slightly different form: instead of having changed the set-point at $t = 0$s (as it has been the case for the generated data that has been used along this chapter), the set-point was changed only as $t = 1000$s. The time window was kept between 0s and 2500s as well as the time interval $\Delta t = 1$s. This only translates the curves 1000s ahead. However, applying the procedure described by Rahman and Choudhury (2011), the canonical correlation vector obtained is now

$$\mathrm{V_{cc}} = \begin{bmatrix} 0.12 \\ 0.53 \\ 0.56 \end{bmatrix}, \tag{4.20}$$

which corresponds to the loop importance index vector

$$\mathrm{LII_{cc}} = \begin{bmatrix} 22.03 \\ 95.06 \\ 100.00 \end{bmatrix}. \tag{4.21}$$

Comparing (4.19) with (4.21), it is easy to see that the same method indicates exactly opposite orders of importance of the loops. It is noteworthy that the authors of the method (Rahman and Choudhury, 2011) concluded that the order was $3 > 2 > 1$ by applying the method to 4000 points (between 0s ans 4000s) with the change of set-point at $t = 1000$s. The order of importance found by them coincides with the one obtained by the method through IAE or ISE ($3 > 2 > 1$) but was achieved in these circumstances. Another test, that consisted on generating data in the same circumstances as those used by Rahman and Choudhury (2011), allowed to reach exactly the same importance order of the loops ($3 > 2 > 1$). Some extra tests were run, from which it was possible to conclude that the

extent of the time window used over the new steady-state does not affect the result. However, the inclusion/exclusion of points from the initial steady-state do affect the results, being able to completely invert the order found by the method. Therefore, the application of this method requires a special case in the establishment of the time-window to be used when collecting the data.

- The second version of this method uses a **canonical correlation matrix** (M_{cc}) instead of a canonical correlation vector. A generic element of this matrix, $M_{cc_{i,j}}$, contains the canonical correlation value between the output variable i and other two output variables when the change in the set-point was applied in loop j. Therefore, the first column contains the canonical correlation values obtained with data for set-point change in loop 1. Within this first column, the first value is the canonical correlation between the controlled variable of loop 1 and the set of controlled variables of the remaining control loops. The second value is the canonical correlation between the controlled variable of loop 2 and the set of controlled variables of the remaining loops (including the controlled variable of loop 1). A similar computing methodology is applied to all columns and rows until completion of the matrix, which is square and has dimension $n_c \times n_c$.

  The computed matrix should then be transformed so that its main diagonal is composed only by unitary values. This transformation is made dividing each row by the diagonal element of that specific row.

  Bryan and Leise (2006) proposed a method to rank web pages based on the connectivity existent among them, called *PageRank* method. Farenzena and Trierweiler (2009) exported this idea to try to rank control loops. Later, Rahman and Choudhury (2010) modified slightly the method proposing to begin with the canonical correlation matrix instead of a special modified partial correlation matrix that was used by Farenzena and Trierweiler (2009). It is this method of Rahman and Choudhury (2010) that is described here.

  According to the method suggested by Rahman and Choudhury (2010), the loop importance index, V_{cc}, is the eigenvector of the transformed matrix M_{cc} which is associated to the eigenvalue closest to 1.

  The last step of the method is equal to that of the first version already described. This step is to rank the loops according to their importance. This is performed by normalizing each element of the vector by the maximum element of that vector and expressing it in

percentage, $LII_{M,cc}$.

The canonical correlation matrix obtained for the data generated in Section 4.1 is

$$M_{cc} = \begin{bmatrix} 0.76 & 0.10 & 0.08 \\ 0.76 & 0.10 & 0.08 \\ 0.24 & 0.14 & 0.08 \end{bmatrix}. \tag{4.22}$$

Matrix M_{cc} is transformed according to the procedure described above to have unitary values in the main diagonal, giving

$$M_{cc,N} = \begin{bmatrix} 1.00 & 0.13 & 0.11 \\ 7.37 & 1.00 & 0.73 \\ 2.97 & 1.77 & 1.00 \end{bmatrix}. \tag{4.23}$$

The eigenvalues (λ_{cc}) of the transformed $M_{cc,N}$ are

$$\lambda_{cc} = \begin{bmatrix} 2.87 + 0.00i \\ 0.06 + 0.21i \\ 0.06 - 0.21i \end{bmatrix} \tag{4.24}$$

and the eigenvectors (Vec_{cc}) are

$$Vec_{cc} = \begin{bmatrix} 0.09 + 0.00i & 0.04 + 0.03i & 0.04 - 0.03i \\ 0.65 + 0.00i & 0.40 - 0.16i & 0.40 + 0.16i \\ 0.76 + 0.00i & -0.90 + 0.00i & -0.90 - 00i \end{bmatrix}. \tag{4.25}$$

The eigenvalue close to 1 found in this case is complex ($\lambda_{cc} = 0.06 + 0.21i$) as well as the corresponding eigenvector

$$Vec_{cc} = \begin{bmatrix} 0.04 + 0.03i \\ 0.40 - 0.16i \\ -0.90 + 0.00i \end{bmatrix}. \tag{4.26}$$

This lacks physical meaning in the context of the problem being solved. Eventually, one can argue that having no physical meaning the pair of complex eigenvalues should be discarded and the third eigenvalue ($\lambda_{cc} = 2.87$) should be selected. The corresponding

eigenvector (the loop importance index) would be

$$\mathrm{Vec_{cc}} = \begin{bmatrix} 0.09 \\ 0.65 \\ 0.76 \end{bmatrix}. \tag{4.27}$$

By normalization between 0 and 100% of the elements of this vector, one would obtain

$$\mathrm{LII_{M,cc}} = \begin{bmatrix} 11.84 \\ 85.53 \\ 100.00 \end{bmatrix}. \tag{4.28}$$

However, the authors of the method do not give any indication how to proceed in the case that the found eigenvalues and eigenvectors are complex numbers. Although the way followed in (4.26), (4.27) and (4.28) is questionable, the truth is that the results obtained following this path coincide with the results obtained by the methods of IAE and ISE, which is the loops to be ranked as $3 > 2 > 1$. The loop importance index vector calculated as described would attribute more importance to loop 3.

Due to the difference of results obtained with the previous version of the canonical correlation method, it was decided to study the inclusion/elimination of points before performing the set-point change. In this context, the same method was applied to the data generated for the special tests of the previous method (see page 49), ie, data between 0s and 2500s having a set-point change performed at $t = 1000$s.

The matrix $\mathrm{M_{cc}}$ obtained is

$$\mathrm{M_{cc}} = \begin{bmatrix} 0.12 & 0.53 & 0.54 \\ 0.25 & 0.53 & 0.14 \\ 0.25 & 0.10 & 0.56 \end{bmatrix}. \tag{4.29}$$

After its transformation, one obtains

$$\mathrm{M_{cc,N}} = \begin{bmatrix} 1.00 & 4.34 & 4.45 \\ 0.48 & 1.00 & 0.27 \\ 0.44 & 0.19 & 1.00 \end{bmatrix}. \tag{4.30}$$

The eigenvalues of the transformed $M_{cc,N}$ are

$$\lambda_{cc} = \begin{bmatrix} 3.13 \\ -0.90 \\ 0.77 \end{bmatrix}. \tag{4.31}$$

The eigenvalue closest to one is the third, to which corresponds the eigenvector (ie, the loop importance index)

$$\mathrm{Vec}_{cc} = \begin{bmatrix} -0.05 \\ -0.71 \\ 0.70 \end{bmatrix}. \tag{4.32}$$

By normalization between 0 and 100% of the elements of this vector, one finally obtains

$$\mathrm{LII}_{M,cc} = \begin{bmatrix} 7.04 \\ 100.00 \\ 98.59 \end{bmatrix}. \tag{4.33}$$

According to this vector, the order of importance of the loops is $2 > 3 > 1$. Once more, the method showed different results just by including or excluding steady-state data before the set-point change is performed. This fact indicates some debilities of the method. Eventually, it needs further refining of its procedure to make it more robust.

IAE technique Other technique to quantify loop interactions was presented by Rossi et al. (2006) and is called **IAE technique**.

The method begins with the application of a normalization factor (NF) in routine data of the process. Such procedure allows the analysis not to be affected by the measurement range. NF can be found from

$$\mathrm{NF} = \min(\mathrm{UCL} - \bar{r}, \quad \bar{r} - \mathrm{LCL}), \tag{4.34}$$

where UCL and LCL are the upper and lower limits of the loop control range,respectively, and $\bar{r}$ is the value of the loop set-point. All controlled variables are divided by their respective normalization factors before subsequent analysis. In this work, it was chosen to define the UCL and LCL values as

$$\mathrm{UCL} = \bar{y} + 3\sigma_y^2 \qquad \text{and} \qquad \mathrm{LCL} = \bar{y} - 3\sigma_y^2. \tag{4.35}$$

To determine the index of interaction, one must first calculate a modified IAE, according to

$$\mathrm{IAE}_k = \mathrm{IAE}_{k-1} + \mid \mathrm{e}_k \mid \Delta t, \tag{4.36}$$

where e_k is the error signal ($\mathrm{e}_k = \mathrm{r}_k - \mathrm{y}_k$ at iteration k) and Δt is the sampling interval. If a change in the error sign occurs, IAE_k is set equal to zero. At the instant when the output variable reaches the set-point value, the IAE value reaches a local maximum value. The comparison of the maximum for different variables can be used to assess the amount of interaction between loops. A change of set-point will generate a local maximum (peak) in the graph IAE in two control loops with significant interaction. As the delay time of each control loop is unknown as well as the delay time in the interaction between the two loops, it is impossible to know the exact time interval in which the interaction between the two loops can be observed. To solve this problem it is necessary to choose a time range $(\mathrm{t_h})$ which includes the two full peaks of the two control loops:

$$\mathrm{t_h} = \mathrm{t_0} + \alpha(\mathrm{t_1} - \mathrm{t_0}), \tag{4.37}$$

where $\mathrm{t_0}$ is the time in which the set-point change starts and $\mathrm{t_1}$ is the time at which the IAE reaches its maximum value (relative to the control loop directly affected by the change). The parameter α is adjustable (Rossi et al., 2006 chose the value 4). The value of this parameter must be greater than the sum of all the delays in the process. To set a time interval to pick up complete changes of all controlled variables.

After estimating these two values of modified IAE, it is possible to assess the interaction via an index. This index is referred to as **Interaction in Time Domain, ITD**. Considering a change in cycle i, the index of interaction, ITD, between i and j loops has the following form

$$\mathrm{ITD}_{i,j} = 1 - \frac{\sum_{\mathrm{t_0}}^{\mathrm{t_h}} \mathrm{IAE}_i}{\sum_{\mathrm{t_0}}^{\mathrm{t_h}} \mathrm{IAE}_i + \sum_{\mathrm{t_0}}^{\mathrm{t_h}} \mathrm{IAE}_j}. \tag{4.38}$$

The method just described in now applied to the generated data (see Section 4.1). First, there is need to define the time to be considered, $\mathrm{t_h}$. For that it is assumed $\mathrm{t_0} = 0$, $\alpha = 4$ while $\mathrm{t_1}$ varies. When the set-point change is made in the first loop (data represented in Figure 4.2), $\mathrm{t_1}$ is set to $\mathrm{t_1} \simeq 800$s (time for which the controlled variable crosses the line of its set-point for the first time). Thus, $\mathrm{t_h} = 3200$s (4.37). In spite of this computed value, it was decided to use $\mathrm{t_h} = 2500$s since the previous generated data was not that long (3200s) and the process looks very stable after 800s.

To calculate the interaction index, ITD, it is necessary to determine the IAE according to

(4.36). Figure 4.7 shows the evolution of modified IAE for each loop.

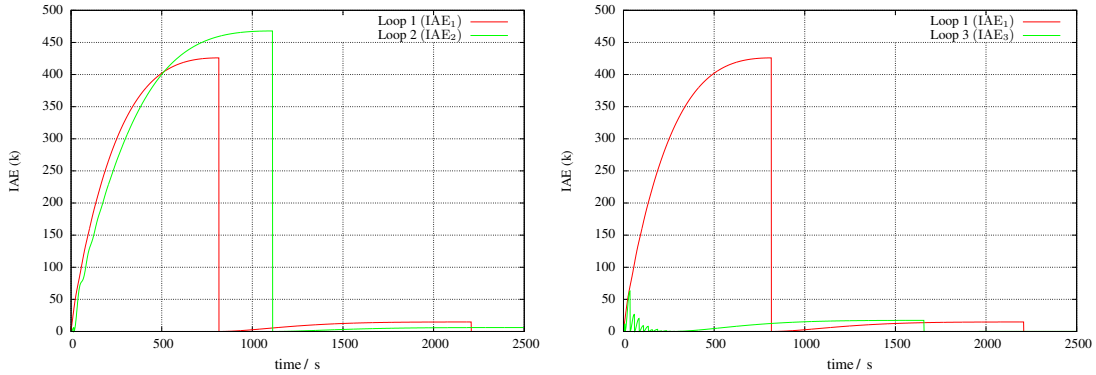


Figure 4.7: Values of modified IAE, after having changed the set-point of loop 1, for loops 1 and 2 and for loops 1 and 3.

After the time interval is set, it is possible to calculate an index that numerically evaluates the interaction between two control loops. Using (4.38), the values of the **Interaction index, ITD**, relative to loop 1 are

$$\mathrm{ITD}_{1,2} = 0.59, \qquad \mathrm{ITD}_{1,3} = 0.06.$$

These values show that the interaction between loop 1 and loop 2 ($\mathrm{ITD}_{1,2} = 0.59$) is much stronger than the interaction between loop 1 and loop 3 ($\mathrm{ITD}_{1,3} = 0.06$). The interaction between loop 1 and loop 2 is considered very significant whereas the interaction between loops 1 and 3 practically does not exist.

For the data corresponding to the set-point change in loop 2 (see Figure 4.3), the first time that the controlled variable reaches the new set-point happens at $t_1 \simeq 40$s.

Thus, $t_h = 0 + 4(40 - 0) = 160$s.

The values of Interaction index, ITD, relative to loop 2 are

$$\mathrm{ITD}_{2,1} = 0.27, \qquad \mathrm{ITD}_{2,3} = 0.22.$$

These values show that loop 2 has significant interactions with the other two loops and that there is no big difference between these two interactions. These values are not comparable with those obtained for the set-point change in loop 1 because the time interval is entirely different.

For the data corresponding to the set-point change in loop 3 (see Figure 4.4), $t_1 \simeq 10$s and therefore, $t_h = 40$s.

Figure 4.8 displays the evolution of the modified IAE over time for each loop.

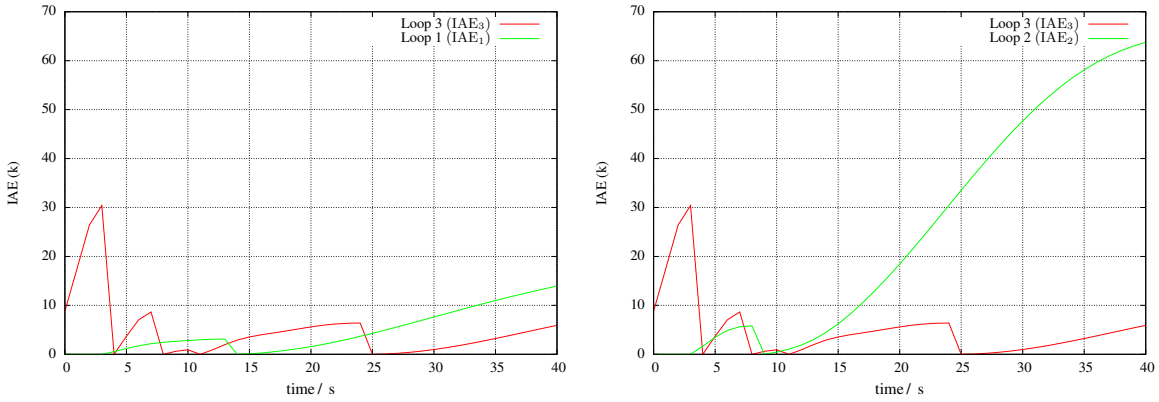


Figure 4.8: Values of modified IAE, after having changed the set-point of loop 3, for loops 3 and 1 and for loops 3 and 2.

The values of the interaction index, ITD, relative to loop 2 are

$$\mathrm{ITD}_{3,1} = 0.47, \qquad \mathrm{ITD}_{3,2} = 0.83.$$

These values show that the interaction between loops 3 and 2 is much higher than the interaction between loops 3 and 1. The interaction between loops 3 and 1 is considered to be strong and the interaction between loops 3 and 2 is considered very strong.

Although Rossi et al. (2006) do not present the results obtained by their method in the form of a matrix, there is, in fact, advantage of using such format, since it eases the comparison between them. In this context, the matrix $\mathrm{M}_{\mathrm{ITD}}$ was built.

All the matrix values were obtained for the same time interval (between 0 and 2500s) resulting

$$\mathrm{M}_{\mathrm{ITD}} = \begin{bmatrix} - & 0.59 & 0.06 \\ 0.85 & - & 0.22 \\ 0.98 & 0.95 & - \end{bmatrix}.$$

This matrix should be analyzed by rows. It ranks the loops as $3 > 2 > 1$. The matrix indicates that loop 3 strongly influences the other two loops but is little influenced by them. Matrix $\mathrm{M}_{\mathrm{ITD}}$ may also be used to see which pair of loops has the greatest interaction. In this case, it is the pair of loops 3 and 1.

***LoopRank* method** The *LoopRank* method, based on *PageRank* algorithm (Bryan and Leise, 2006), was developed by Farenzena and Trierweiler (2009) and uses the partial correlation between the output variables of each control loop to assess the interactions. The first step of the method is to calculate the matrix of the partial correlations A_{LR}, that works as a relative weight

matrix. The matrix element of row i and column j is the partial correlation between the variable of loop i and the variable of loop j. Each value of this matrix provides the relative weight of loop j with respect to loop i. The next step is to use *PageRank* algorithm on matrix A_{LR} to order the loops. However, matrix A_{LR} in *PageRank* algorithm was defined as having null diagonal elements. Having now defined $A_{LR}{}'$ by the partial correlations, its diagonal is constituted by 1. Before continuing Farenzena and Trierweiler (2009) set the diagonal of matrix $A_{LR}{}'$ from 1 to 0. The *PageRank* algorithm quantifies the relative importance of each loop using the importance score vector, IS. To calculate this vector, one must first carry out the SVD of matrix A_{LR} with zeros in the main diagonal, that is,

$$A_{LR} = U\Sigma V^{T}. \tag{4.39}$$

The importance score vector is the left singular vector of this matrix A_{LR} (in matrix U) which is associated to the singular value (matrix Σ) closest to one. To rank the importance between 0 and 100%, this vector, **LoopRank vector**, is normalized so that the sum of all elements is 100%. The control loop with higher value in this vector is the one with more interaction and it is the one that should be given priority in maintenance.

The *LoopRank* method was implemented in `Octave` and tested with the previously generated data (see Section 4.1. The dataset is constituted by the three subsets (each obtained by changing the set-point of a loop). The corresponding matrix of the partial correlations is

$$A_{LR}{}' = \begin{bmatrix} 1.00 & 0.85 & -0.08 \\ 0.85 & 1.00 & 0.35 \\ -0.08 & 0.35 & 1.00 \end{bmatrix}. \tag{4.40}$$

As expected, (4.40) has unit values on its main diagonal. Therefore, (4.40) is updated by replacing those values with zeros. The new matrix A_{LR} is

$$A_{LR} = \begin{bmatrix} 0.00 & 0.85 & -0.08 \\ 0.85 & 0.00 & 0.35 \\ -0.08 & 0.35 & 0.00 \end{bmatrix}. \tag{4.41}$$

The decomposition SVD of matrix (4.41) is

$$A_{LR} = \begin{bmatrix} 0.65 & 0.66 & -0.38 \\ -0.69 & 0.72 & 0.06 \\ 0.31 & 0.22 & 0.92 \end{bmatrix} \times \begin{bmatrix} 0.95 & 0.00 & 0.00 \\ 0.00 & 0.89 & 0.00 \\ 0.00 & 0.00 & 0.06 \end{bmatrix} \times \begin{bmatrix} -0.65 & 0.66 & -0.38 \\ 0.69 & 0.72 & 0.06 \\ -0.31 & 0.22 & 0.92 \end{bmatrix}. \tag{4.42}$$

From (4.42), it is possible to see that the singular value closest to 1 (which is 0.95) corresponds to the singular vector constituted by the first column of U, that is,

$$\mathrm{IS} = \begin{bmatrix} 0.65 \\ -0.69 \\ 0.31 \end{bmatrix}. \tag{4.43}$$

Normalizing IS, one obtains the *LoopRank* vector, LR,

$$\mathrm{LR} = \begin{bmatrix} 39 \\ 42 \\ 19 \end{bmatrix}. \tag{4.44}$$

This method attributes bigger importance to loop 2, followed by loop 1 and, finally, by loop 3. This ranking is different from the ranking obtained by the other analysed methods. Therefore further testing was performed with another dataset that differs from the previous dataset just in the moment of occuring the set-point change (now at $t = 1000$s). The time window of the dataset is between 0s and 2500s, corresponding to 2500 points.

The resulting matrix of partial correlations is

$$\mathrm{A_{LR}}' = \begin{bmatrix} 1.00 & 0.90 & -0.84 \\ 0.90 & 1.00 & 0.88 \\ -0.84 & 0.88 & 1.00 \end{bmatrix}, \tag{4.45}$$

or, after forcing the diagonal to be zero,

$$\mathrm{A_{LR}} = \begin{bmatrix} 0.00 & 0.90 & -0.84 \\ 0.90 & 0.00 & 0.88 \\ -0.84 & 0.88 & 0.00 \end{bmatrix}. \tag{4.46}$$

After an SVD, the importance score vector (IS), corresponding in this case to a singular value of 0.91, is

$$\mathrm{IS} = \begin{bmatrix} 0.52 \\ 0.80 \\ 0.29 \end{bmatrix}. \tag{4.47}$$

The *LoopRank* vector is calculated by normalizing IS, obtaining

$$\mathrm{LR} = \begin{bmatrix} 32 \\ 50 \\ 18 \end{bmatrix}. \tag{4.48}$$

The method gave more importance to loop 2, then to loop 1 and less importance to loop 3. It is clear to see that the results do not capture the real importance of the loops.

It is noteworthy that the loop ranking based on the normalized importance score vector considers loop interaction only. However, other criteria, such as impact on plant profitability or in the cost of operations may be preferred (Farenzena and Trierweiler, 2009). In this case, it is important to combine the obtained interaction results with other criteria that define the overall importance of each control loop on the plant. A vector, designated by **LoopRank** is defined with weights w_i, that may depend on the type of loop and on the profitability. Farenzena and Trierweiler (2009) suggest that flow and level loops are the least important ones ($\mathrm{w}_i = 1$), pressure loops have middle importance ($\mathrm{w}_i = 1.5$) and temperature and composition are of the highest importance ($\mathrm{w}_i = 2$). To calculate this *LoopRank*, it is necessary to multiply each element of the vector by the corresponding loop weight (w_i). After normalization, it is obtained the vector

$$\mathrm{LR} = \begin{bmatrix} \mathrm{w}_1 \mathrm{IS}_{\mathrm{N},1} \\ \vdots \\ \mathrm{w}_\mathrm{n} \mathrm{IS}_{\mathrm{N,n}} \end{bmatrix}. \tag{4.49}$$

Variability matrix Farenzena et al. (2009) proposed a method of quantification of loop interaction that is based on the **Variability Matrix, VM**.

The first step of the method consists on the calculation of the actual variance of the controlled variable of each control loop ($\sigma_{\mathrm{y},i}$ where $i = 1, \cdots, \mathrm{n_c}$). The next step is to determine the optimal controller for each control loop, to apply this new controller, one at a time and record the variances of the output variables. The last step is to build the variability matrix, VM. Each element of this matrix is calculated according to

$$\mathrm{VM}(i,j) = \frac{\sigma^2_{\mathrm{y}_i} - \sigma^2_{\mathrm{y}_{i,j}}}{\sigma^2_{\mathrm{y}_i}}, \tag{4.50}$$

where $\sigma^2_{\mathrm{y}_i}$ is the current variance of the output variable i and $\sigma^2_{\mathrm{y}_{i,j}}$ is the variance of the output variable i when the controller of loop j is changed.

The analysis of the matrix of variability is performed by columns. Each value represents the relative amount of variance of the output variable of loop i that may vary changing the controller of loop j. The loop with greater interaction is one that has a column that has the largest values relative to the value in its diagonal.

This method of analysis of interactions between control loops can also be combined with an economic analysis. This combination is done through a relationship between improved performance and economic benefits, given by vector $\mathrm{D}_\diamondsuit$, that quantifies economically the variability reduction per unit time ($\mathrm{n_c}$ is the number of main loops), one can write

$$\mathrm{D}_\diamondsuit = [\mathrm{D}_{\diamondsuit,1}, \mathrm{D}_{\diamondsuit,2}, \cdots, \mathrm{D}_{\diamondsuit,\mathrm{n_c}}]. \tag{4.51}$$

The elements of vector $\mathrm{D}_\diamondsuit$ can be provided by the sales department or by the teams responsible for the control of the company at the highest level. The economic benefit can be quantified using the simple relationship

$$\mathrm{CLEB} = \mathrm{D}_\diamondsuit \times \mathrm{VM}, \tag{4.52}$$

where **CLEB** is the **Control Loop Economic Benefit vector**. This vector is easier to analyze than matrix VM. In the present study the prioritization of control loops was found not taking into account the economic perspective. Therefore, $\mathrm{D}_\diamondsuit$ is considered to be a unitary row vector $\mathrm{D}_\diamondsuit = [1, 1, \cdots, 1]$.

For the sake of application of this method, the simulated closed-loop data was obtained with simple PID controllers whose transfer functions are

$$\mathrm{G_c} = \begin{bmatrix} \frac{25s+0.25}{100s} & 0 & 0 \\ 0 & \frac{65s+0.65}{100s} & 0 \\ 0 & 0 & \frac{5s+1.5}{10s} \end{bmatrix}. \tag{4.53}$$

Since the subject of this study is loop interactions, no special efforts on tuning of these controllers were done. The method was applied to closed-loop data without set-point changes. A source of white noise with unit variance was introduced to improve the information contained in the data. The multivariable disturbance model (Farenzena et al., 2009) is defined as

$$\mathrm{G_\varepsilon} = \begin{bmatrix} \frac{1.2}{45s+1}e^{-27s} & \frac{1.44}{40s+1}e^{-27s} \\ \frac{1.52}{25s+1}e^{-15s} & \frac{1.83}{20s+1}e^{-15s} \\ \frac{1.14}{27s+1} & \frac{1.26}{32s+1} \end{bmatrix}. \tag{4.54}$$

Figure 4.9 shows closed-loop simulation results.

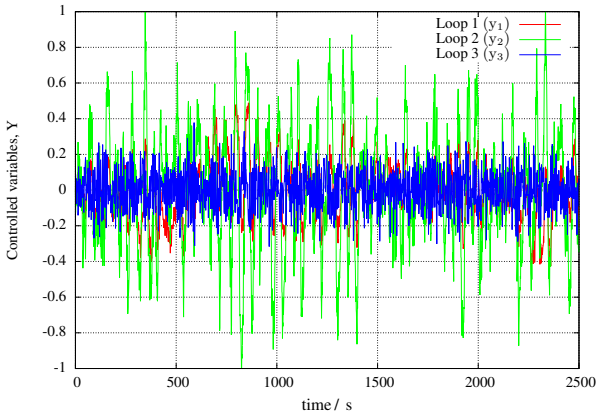


Figure 4.9: Output variable profiles with white noise (with unit variance) and without changes of set-point (first dataset).

The output variable variances under PID control are given in Table 4.1. Two simulations were performed with the same conditions.

Table 4.1: Output variable variance in closed-loop.

y_i	σ^2_{1,y_i}	σ^2_{2,y_i}
1	3.24×10^{-02}	3.04×10^{-02}
2	1.47×10^{-01}	1.37×10^{-01}
3	1.65×10^{-02}	1.61×10^{-02}

The variable with the greatest variability belongs to loop 2 and the lowest variability is in loop 3. The following step is the determination of the new controller for each control loop and the recording of the variances of the three output variables. The new controllers should be optimal controllers for each loop. However, the controllers used continue to be simple and only change the variance of the output variable of this control loop. In this study, the new controllers of loops 1 and 2 cause a decrease in variance of its output variables. The controller of loop 3 increases the variance of its output variable.

The first change is made in the controller of loop 1 $(G_{c,11})$ and its transfer function becomes

$$G_{c,11} = \frac{5s + 0.25}{100s}. \tag{4.55}$$

Figure 4.10 illustrates the simulated results with this new controller.

The output variables variances, with the new controller (4.55), are summarized in Table 4.2.

Note that all variances decrease. However, the decrease was greater in loop 1 (where the controller has changed). The reduction of the variance influences more loop 3 than loop 2.

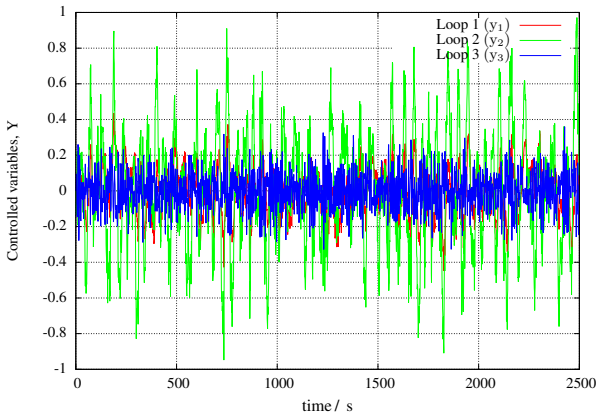


Figure 4.10: Output variable profiles with the new loop 1 controller.

Table 4.2: Output variables variances with the new loop 1 controller.

y_i	$\sigma^2_{1,y_i,1}$	$\sigma^2_{2,y_i,1}$
1	2.76×10^{-02}	2.28×10^{-02}
2	1.37×10^{-01}	1.21×10^{-01}
3	1.49×10^{-02}	1.40×10^{-02}

The second change is made in the controller of loop 2 $(G_{c,22})$. Its updated structure is given by

$$G_{c,22} = \frac{10s + 0.65}{100s}. \tag{4.56}$$

Figure 4.11 shows the simulated results with this new controller.

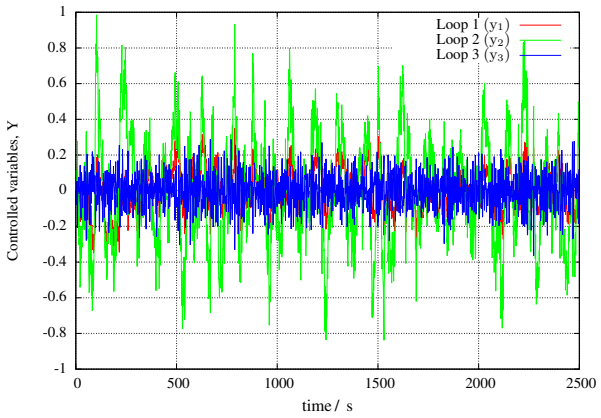


Figure 4.11: Output variable profiles with the new loop 2 controller.

The values of the variabilities of the three output variables with this new controller (4.56) are presented in Table 4.3.

There was a reduction in variability, being this decrease less pronounced in loop 3.

Table 4.3: Output variable variances with the new loop 2 controller.

y_i	$\sigma^2_{1,y_i,2}$	$\sigma^2_{2,y_i,2}$
1	1.60×10^{-02}	1.70×10^{-02}
2	8.59×10^{-02}	9.10×10^{-02}
3	1.41×10^{-02}	1.45×10^{-02}

The last change is made in the controller of loop 3 ($G_{c,33}$), whose structure becomes

$$G_{c,33} = \frac{15s + 1.5}{10s}, \tag{4.57}$$

and the corresponding closed-loop simulation results are presented in Figure 4.12. Table 4.4

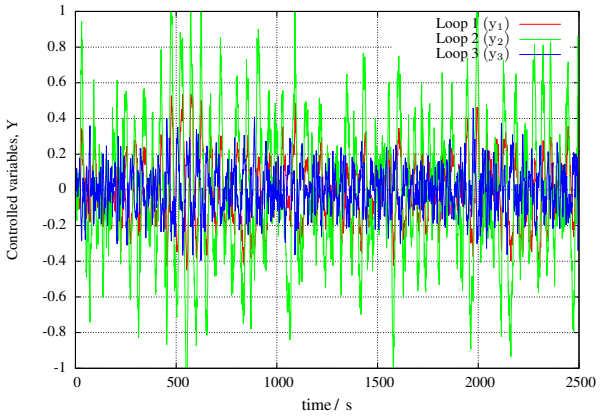


Figure 4.12: Output variable profiles with the new loop 3 controller.

summarizes the variability of output variables with this new controller.

Table 4.4: Output variable variances with the new loop 3 controller.

y_i	$\sigma^2_{1,y_i,3}$	$\sigma^2_{2,y_i,3}$
1	1.81×10^{-02}	1.89×10^{-02}
2	1.01×10^{-01}	1.02×10^{-01}
3	1.06×10^{-02}	1.08×10^{-02}

Clearly, all output variables present a significant variability decrease. Based on the results of the tests with controller changes, the variability matrix is built as

$$VM_1 = \begin{bmatrix} 0.15 & 0.51 & 0.44 \\ 0.07 & 0.41 & 0.31 \\ 0.10 & 0.14 & 0.36 \end{bmatrix}, \qquad VM_2 = \begin{bmatrix} 0.25 & 0.44 & 0.38 \\ 0.11 & 0.33 & 0.26 \\ 0.13 & 0.10 & 0.33 \end{bmatrix}. \tag{4.58}$$

Matrix VM_1 (4.58) may be interpreted as follows. The first column indicates that when the

variance of loop 1 decrease 15%, the variance of loop 2 decreases 7% and the variance of loop 3 decreases 10%. In the second column, when the variance of loop 2 decrease 41%, the variance of loop 1 decreases 51% and the variance of loop 3 decreases 14%. In the third column, when the variance of loop 3 decrease 36%, the variance of loop 1 decreases 44% and the variance of loop 2 decreases 31%. A similar interpretation may be applied to matrix VM_2 corresponding to the second dataset.

These two matrices clearly show that loop 1 has less influence that the other two loops. Between these two loops, the major difference is in the the relationship between themselves. The results show that loop 3 affects more loop 2 than unlike. The decrease of the variability caused by loop 2 in loop 3 is less than half of the its diagonal value while the decrease of the variability caused by loop 3 in loop 2 is significantly higher. According to this method, the loops can be ranked as 3 > 2 > 1.

As already mentioned, this method can be combined with economic aspects with the variability matrix being transformed into a vector (**Control Loop Economic Benefit vector, CLEB**), to help the analysis of the prioritization of control cycles. In this study, it was considered that the prioritization of control actions is only due to the interactions. Therefore, $D_{\diamondsuit} = [1, 1, 1]$. By applying (4.52), the two control loop economic benefit vectors are

$$\text{CLEB} = [0.32, 1.06, 1.10] \quad \text{and} \quad \text{CLEB} = [0.49, 0.87, 0.97]. \tag{4.59}$$

As can be immediately noted, the results are the same as those observed with the variability matrices, being easier to look at the vector. These latter results show a great similarity between loops 2 and 3. However, this similarity is not that pronounced in the variability matrix. The analysis of this method showed that it is necessary to have some caution in this last step (4.52), since, if there are significant differences in the values of the main diagonal of the variability matrix, the values of the vector CLEB will be affected. The loops with higher values on the main diagonal will be benefited. This problem does not arise if a proper assessment is made in the matrix, ie, the values presented in matrix should always be analyzed comparing the value with the value of the main diagonal.

After obtaining the results of the various methods, they were compared and it was observed that some methods originate different orders, indicating potential failures of some methods. The methods using IAE or ISE, IAE technique and the variability matrix obtained results that are consistent between themselves and also seem to be in accordance with a visual analysis of the data. The ranking of loops is 3 > 2 > 1. The methods that used the canonical correlation and the

LoopRank suggested different results.

4.3 Brief Analysis with Systems B and C

To further evaluate the results presented in this section, the same methods were applied to data of different processes, namely processes B and C (see Section 4.1 for details on the generation of the datasets from these two systems).

In Table 4.5 are summarized the results obtained using the methods of prioritization of control loops for processes $G_{p,B}$ and $G_{p,C}$ as well as the results of the process that was used to provide a detailed explanation of each of the methods, $G_{p,A}$.

Table 4.5: Results obtained by various methods to the simulated data of three different processes.

Method	$G_{p,A}$	$G_{p,B}$	$G_{p,C}$
Expected ranking	$3 > 2 > 1$	$1 > 2 > 3$	$1 > 2 > 3$
IAE	$[2.2, 7.0, 100.0]^T$ √	$[100.0, 48.2, 47.9]^T$ √	$[100.0, 49.8, 50.3]^T$ x
ISE	$[1.2, 4.4, 100.0]^T$ √	$[100.0, 56.4, 55.7]^T$ √	$[100.0, 61.4, 60.5]^T$ √
LII_{cc}	$[100.0, 13.6, 10.7]^T$ x	$[75.0, 100.0, 97.2]^T$ x	$[100.0, 32.6, 65.1]^T$ x
$LII_{M,cc}$	$[11.8, 85.5, 100.0]^T$ √	$[83.9, 93.5, 100.0]^T$ x	$[100.0, 91.7, 96.7]^T$ x
LoopRank	$[39, 42, 19]^T$ x	$[5, 50, 45]^T$ x	$[4, 50, 46]^T$ x
IAE technique	$\begin{bmatrix} - & -0.59 & 0.06 \\ 0.85 & - & 0.22 \\ 0.98 & 0.95 & - \end{bmatrix}$ √	$\begin{bmatrix} - & 0.14 & 0.16 \\ 0.07 & - & 0.05 \\ 0.32 & 0.06 & - \end{bmatrix}$ x	$\begin{bmatrix} - & 0.05 & 0.06 \\ 0.04 & - & 0.02 \\ 0.18 & 0.02 & - \end{bmatrix}$ x
VM	$\begin{bmatrix} 0.15 & 0.51 & 0.44 \\ 0.07 & 0.41 & 0.31 \\ 0.10 & 0.14 & 0.36 \end{bmatrix}$ √	$\begin{bmatrix} -1.16 & -0.21 & 0.01 \\ -0.08 & -1.36 & -0.03 \\ -0.67 & -0.19 & -0.60 \end{bmatrix}$ √	$\begin{bmatrix} -0.29 & -0.27 & 0.02 \\ -0.09 & -1.09 & 0.02 \\ -0.26 & -0.20 & -0.51 \end{bmatrix}$ √
CLEB	$[0.32, 1.06, 1.10]^T$ √	$[-1.92, -1.77, -0.62]^T$ √	$[-0.64, -1.56, -0.46]^T$ x

For the total of the three processes generated data, only one method did always suggest the expected rank, **using ISE**. It is worth mentioning that the expected rank, in case 1, is the one that seems more appropriate from a simple visual analysis of the data (comparing the output variables with their set-points changes) and, thus, is a rough measure of the error, as IAE and ISE do, although in a much less rigorous way.

In spite of not always suggesting what was considered as "expected rank", there are two other methods that can be classified as giving good results. One uses **IAE** and the other the **variability matrix**. The method **using IAE** do not suggest the expected rank only with data of process $G_{p,C}$. However, the values obtained for loops 2 and 3 are very similar, what indeed is possible to detect by visual analysis of the data (see Figure 4.6). This similarity of values reveals that the apparent failure of the method in this situation is not that important on even it is not a failure. In the method that uses the matrix of variability, the results of the computed matrix looks convincing. However, after calculating the vector CLEB, the method seems alter the direction and suggests a different loop rank, as it can be seen with the results obtained for the data of process $G_{p,C}$. This happens because in the step of multiplying vector $D_{\diamondsuit}$ by matrix VM to obtain the vector CLEB, that supposedly would give an indication of the importance of the loops, the effect of a certain loop on itself is also being accounted, distorting an analysis of pure interaction between loops. This undesirable effect could be eliminated by a slight change of the method consisting on a normalization by columns, in order to have always the same value in the diagonal of VM.

The method of the variability matrix was still tested using simulated data of the processes $G_{p,B}$ and $G_{p,C}$ when there is no noise present and there are only three changes of set points. Table 4.6 exposes the corresponding results together with the results obtained with a second dataset of the original process.

The results presented confirm what has been concluded from the results shown in Table 4.5, ie, this method works well only when the matrix is analyzed as an end result.

The other methods, the *LoopRank* method, the methods that use canonical correlation and the IAE technique, do not show consistent results and therefore should be reviewed.

Note that during the application of the **technique IAE**, it was found that the suggested rank did not coincide with the expected rank using data from processes $G_{p,B}$ and $G_{p,C}$ because the values of the output variables rather than oscillating around the set point will present a very

Table 4.6: Results obtained with new data through the variability matrix method for the three processes.

Method	$G_{p,A}$	$G_{p,B}$	$G_{p,C}$
Expected ranking	3 > 2 > 1	1 > 2 > 3	1 > 2 > 3
VM	$\begin{bmatrix} 0.25 & 0.44 & 0.38 \\ 0.11 & 0.33 & 0.26 \\ 0.13 & 0.10 & 0.33 \end{bmatrix}$ $\surd$	$\begin{bmatrix} -0.53 & -0.02 & 0.00 \\ -0.38 & -0.61 & 0.00 \\ -0.35 & -0.01 & -0.36 \end{bmatrix}$ $\surd$	$\begin{bmatrix} -0.94 & -0.05 & 0.00 \\ -0.32 & -0.87 & 0.00 \\ -0.37 & -0.09 & -0.83 \end{bmatrix}$ $\surd$
CLEB	$[0.49, 0.87, 0.97]$ $\surd$	$[-1.26, -0.64, -0.36]$ $\surd$	$[-1.63, -1.01, -0.83]$ $\surd$

slow convergence when they are very close the set-point. According to this method, in that circumstances the modified IAE value is always increasing.

Chapter 5

Conclusions and Future Work

The work here developed is composed of two main parts. The first is concerned with the development of a tool to assess and monitor the performance of multivariable control systems, which presents a series of appealing characteristics such as to be based only on industry data and not on models, to be susceptible of on-line application, not requiring knowledge of dynamic models of processes and to be potentially applicable to all kinds of time invariant processes. However, it has two significant drawbacks, the need to introduce a source of undesirable disturbance in the process (dither signal) and the computational problems that some datasets induce. However, in the case of problematic datasets it is always possible to try to use the method applied to a different dataset of the same system because this matrix depends only on the dynamic models of the process. Therefore, if the process has not changed, this matrix is the same and it is not necessary to determine it for all the datasets.

If the dynamic process models are known, the interactor matrix can be computed directly by Rogozinski's algorithm, which was implemented in `Octave`. The program was tested and used.

However, in the industrial practice it is more common to resort to algorithms that use process data instead of explicit dynamic models. This process data is used to determine the initial conditions of the Rogozinski's algorithm. This algorithm was implemented and tested as part of the developed tool.

This work also studied the influence of process gains on the interactor matrix, a subject that had not been previously reported in the literature. It was found out that changes of process gains do not alter the structure of the matrix and that the higher sensitivity is associated to the submodels with the lowest delay. When these gains were altered in submodels with the greatest delay, the interaction matrix remained the same.

Future research works should address systems with higher number of output variables and delays.

The use of a dither signal improves the accuracy of the interactor matrix estimation, at the expense of bigger disturbances in the process.

The developed tool was tested with two sets of simulated data maintaining the same process and the same controller, changing only the external disturbances models ($G_{\varepsilon,A}$ and $G_{\varepsilon,B}$). Obtained results are consistent with those found in the literature (Huang, 1997 and Jelali, 2010) and the control system with the model $G_{\varepsilon,B}$ presented better performance showing that performance depends not only of the process but also on external factors.

Besides, the results of this study showed that the performance of a control system is impacted by the interaction among control loops.

In the case of significant interaction, methods that use the determinant of the covariances matrix are more adequate than those that use the trace overlooking the off-diagonal elements that are due to the interaction.

As a possible solution, a poorly performing controller may be retuned or replaced with a more adequate one.

Most performance metrics account for process variability only. One of possible future work directions is to consider and to optimize the control action costs.

The developed tool includes a method that determines the upper and the lower limit of the performance. This method does not use the interactor matrix and is especially indicated where it is difficult to obtain this matrix. The results were consistent with those obtained using the interaction matrix.

One of the objectives of this study was to implemented and test methods for loop maintenance prioritization that takes into account the interactions among control loops.

The analysed methods presented weaknesses and need further work work to comprove their robustness. Particularly, the *LoopRank*, the canonical correlation methods and the IAE technique.

Besides, all the methods with the exception of that based on the variability matrix, consider the controlled variables only.

In these methods, one must be careful because a loop with a more oscillatory behavior can amplify the influence of the other loops. When performing a disturbance in another loop, if this fluctuates a lot, a small interaction serves to oscillate a lot this and conclude that the interaction is high.

The results should be presented in the form of variability matrix or the vector CLEB needs

to be normalized by diagonal value of this matrix.

The other methods do not show consistent results and therefore should be reviewed.

Note that the technique IAE only works well when the response of the controlled variables oscillate between the value of the set point and not when the variables not oscillate and take a long time to reach the set-point.

Nomenclature

Symbol	Description
A_{LR}	Relative weight matrix for *LoopRank* method
A_{LR}'	Partial correlation matrix for *LoopRank* method
AC (k)	Auto-correlation
CCV	Canonical correlation value
cc	Canonical correlation
D	Interactor matrix
D_w	Weighted unitary interactor matrix
$D_\diamondsuit$	Vector that quantifies economically the variability reduction
d	Dither signal
e	Control error
G°	Block - Toeplitz Matrix
G_c	Controller transfer function
G_d	High-pass filter for dither signal
G_m	Sensor transfer function
G_p	Process transfer function
$G_{p,cl}$	Closed loop process transfer function
$\tilde{G}_p$	Delay free process transfer function
G_ε	Disturbance transfer function
I	Identity matrix
I_v	Volume ratio of the ellipsoids (performance index)
IS	Importance score vector
ITD	Interaction in time domain index
LCL	Lower control limit
LI	Loop interaction index

LII	Loop importance index
LR	*LoopRank* vector
l	Number of principal components
M	Loop interaction matrix
M_{cc}	Canonical correlation matrix
N	Matrix with the polynomial of the numerator of process transfer function
N_a	Data lenght
NF	Normalization factor
n	Number of infinite zeros
n_c	Number of the control loops
R	Matrix with the polynomial of the denominator of process transfer function
R(i)	Cross-correlation
r	Set-point
U	Manipulated vaiables (MIMO systems)
u	Manipulated variable or control signal
Y	Controlled variables (MIMO systems)
$\tilde{Y}$	Interactor-filtered variable
y	Controlled or output variable
W	Weighting matrix
U	Row shift polynomial matrix
UCL	Upper control limit
V_{cc}	Canonical correlation vector
Vec	Eigenvector

Nomenclature - Greek

Symbol	Description
ε	External disturbance variable
η	Performance index
θ	Delay of the process or order of the interactor matrix
Γ	Unitary real matrix

Λ	Block-matrix of the first θ Markov parameters
λ	Eigenvalue
ρ	Partial correlation
Σ_y	Covariance matrix of the real process
Σ_{mv}	Covariance matrix with minimum variance controller
σ^2	variance

Acronyms

Acronym	Description
ARX	Auto-regressive model
CLEB	Control loop economic benefit vector
cov	Covariance matrix
CPA	Control performance assessment
CPM	Control performance monitoring
det	Determinant
FCOR	Filtering and correlation algorithm
GMV	Generalized minimum variance
IAE	Integral of absolute error
IR	Impulse response
ISE	Integral of squared error
LQ	Linear quadractic
LQG	Linear-quadractic gaussian
MIMO	Multivariable input multivariable output
MPC	Model predictive control
MV	Minimum variance
MVC	Minimum variance controller
PCA	Principal components analysis
PCS	Principal component subspace
PID	Proportional integral derivative controller
RMFD	Right matrix fraction description
SVD	Singular value decomposition
VM	Variability matrix

Bibliography

Aström, K. (1970). *Introduction to stochastic control theory*. Mathematics in science and engineering. Academic Press.

Box, G., Jenkins, G., and Reinsel, G. (1970). *Time series analysis: forecasting and control*. Forecasting and Control Series. Prentice Hall.

Bryan, K. and Leise, T. (2006). The $25,000,000,000 Eigenvector: The Linear Algebra behind Google. *SIAM REVIEW - Society for Industrial and Applied Mathematics*, 48(3):569 – 581.

Desborough, L. and Harris, T. (1992). Performance assessment measures for univariate feedback control. *The Canadian Journal of Chemical Engineering*, 70(6):1186–1197.

Desborough, L. and Harris, T. (1993). Performance assessment measures for univariate feedforward/feedback control. *The Canadian Journal of Chemical Engineering*, 71(4):605–616.

DeVries, W. and Wu, S. (1978). Evaluation of process control effectiveness and diagnosis of variation in paper basis weight via multivariate time-series analysis. *Automatic Control, IEEE Transactions on*, 23(4):702 – 708.

Eriksson, P.-G. and Isaksson, A. (1994). Some aspects of control loop performance monitoring. In *Control Applications, 1994., Proceedings of the Third IEEE Conference on*, pages 1029 –1034 vol.2, Glasgow.

Ettaleb, L. (1999). *Control loop performance assessment and oscillation detection*. PhD thesis, University of British Columbia, Vancouver, Canada.

Farenzena, M. and Trierweiler, J. O. (2009). LoopRank: A Novel Tool to Evaluate Loop Connectivity. In Engell, Sebastian, A., editor, *7th IFAC International Symposium on Advanced Control of Chemical Processes, 2009*, pages 964 – 969, Koç University, Turkey.

Farenzena, M., Trierweiler, J. O., and Shah, S. L. (2009). Variability Matrix: A Novel Tool to Prioritize Loop Maintenance. In Engell, Sebastian, A., editor, *7th IFAC International Symposium on Advanced Control of Chemical Processes, 2009*, pages 673 – 678, Koç University, Turkey.

Gao, J., Patwardhan, R., Akamatsu, K., Hashimoto, Y., Emoto, G., Shah, S. L., and Huang, B. (2003). Performance evaluation of two industrial MPC controllers. *Control Engineering Practice*, 11(12):1371 – 1387. <ce:title>Award winning applications-2002 {IFAC} World Congress</ce:title>.

Goodwin, G. C., Graebe, S. F., and Salgado, M. E. (2001). *Control System Design*. Prentice Hall.

Grimble, M. (2002). Controller performance benchmarking and tuning using generalised minimum variance control. *Automatica*, 38(12):2111 – 2119.

Harris, T., Boudreau, F., and Macgregor, J. (1996). Performance assessment of multivariable feedback controllers. *Automatica*, 32(11):1505 – 1518.

Harris, T. J. (1989). Assessment of Control Loop Performance. *The Canadian Journal of Chemical Engineering*, 67:856–861.

Horch, A. and Isaksson, A. J. (1999). A modified index for control performance assessment. *Journal of Process Control*, 9(6):475 – 483.

Hovd, M. (2009). Lecture notes for the course Advanced Control of Industrial Processes. Technical report, Institutt for Teknisk Kybernetikk, NTNU.

Huang, B. (1997). *Multivariate Statistical Methods For Control Loop Performance Assessment*. PhD thesis, Department of Chemical Engineering, Faculty of Graduate Studies and Research, University of Alberta, Edmonton, Alberta.

Huang, B. (2002). Minimum variance control and performance assessment of time-variant processes. *Journal of Process Control*, 12(6):707 – 719.

Huang, B., Ding, S. X., and Qin, S. (2005a). Closed-loop subspace identification: an orthogonal projection approach. *Journal of Process Control*, 15(1):53 – 66.

Huang, B., Ding, S. X., and Thornhill, N. (2005b). Practical solutions to multivariate feedback control performance assessment problem: reduced a priori knowledge of interactor matrices. *Journal of Process Control*, 15(5):573 – 583.

Huang, B. and Shah, S. (1999). *Performance Assessment of Control Loops: Theory and Applications*. Advances in Industrial Control. Springer.

Huang, B., Shah, S., and Kwok, E. (1997a). Good, bad or optimal? performance assessment of multivariable processes. *Automatica*, 33(6):1175 – 1183.

Huang, B., Shah, S., and Miller, R. (2000). Feedforward plus feedback controller performance assessment of mimo systems. *Control Systems Technology, IEEE Transactions on*, 8(3):580 –587.

Huang, B. and Shah, S. L. (1998). Practical issues in multivariable feedback control performance assessment. *Journal of Process Control*, 8(5-6):421 – 430. <ce:title>ADCHEM '97 {IFAC} Symposium: Advanced Control of Chemical Processes</ce:title>.

Huang, B., Shah, S. L., Kwok, K. E., and Zurcher, J. (1997b). Performance assessment of multivariate control loops on a paper-machine headbox. *The Canadian Journal of Chemical Engineering*, 75(1):134–142.

Jelali, M. (2010). *Control system performance monitoring – Assessment, diagnosis and improvements of control loop performance in industrial automation*. PhD thesis, University of Duisburg-Essen, Essen, Germany.

Julien, R. H., Foley, M. W., and Cluett, W. R. (2004). Performance assessment using a model predictive control benchmark. *Journal of Process Control*, 14(4):441 – 456.

Ko, B.-S. and Edgar, T. (1998). Assessment of achievable pi control performance for linear processes with dead time. In *American Control Conference, 1998. Proceedings of the 1998*, volume 3, pages 1548 –1552 vol.3, Philadelphia.

Ko, B.-S. and Edgar, T. F. (2000). Performance assessment of cascade control loops. *AIChE Journal*, 46(2):281–291.

Ko, B.-S. and Edgar, T. F. (2001a). Performance assessment of constrained model predictive control systems. *AIChE Journal*, 47(6):1363–1371.

Ko, B.-S. and Edgar, T. F. (2001b). Performance assessment of multivariable feedback control systems. *Automatica*, 37(6):899 – 905.

Kozub, D. and Garcia, C. (1993). Monitoring and diagnosis of automated controllers in the chemical process. In *Proc AIChE*, St. Louis-USA.

Majecki, P. and Grimble, M. (2004a). Controller performance design and assessment using nonlinear generalized minimum variance benchmark: scalar case. In *Proc. Control 2004*. University of Bath, UK.

Majecki, P. and Grimble, M. J. (2004b). Gmv and restricted - structure gmv controller performance assessment - multivariable case. In *Proceedings of the 2004 American Control Conference*, volume 1, pages 697 – 702.

McNabb, C. A. and Qin, S. (2003). Projection based mimo control performance monitoring: I – covariance monitoring in state space. *Journal of Process Control*, 13(8):739 – 757.

McNabb, C. A. and Qin, S. (2005). Projection based MIMO control performance monitoring: II - measured disturbances and setpoint changes. *Journal of Process Control*, 15(1):89 – 102.

Olaleye, F., Huang, B., and Tamayo, E. (2004). Performance assessment of control loops with time-variant disturbance dynamics. *Journal of Process Control*, 14(8):867 – 877.

Patwardhan, R. (1999). *Studies in Synthesis and Analysis of Model Predictive Controllers*. PhD thesis, University of Alberta.

Patwardhan, R. S., Shah, S., Emoto, G., and Fujii, G. (1998). Performance analysis of model-based predictive controllers: an industrial study. In *Proc. AIChE*, Miami - USA.

Paulonis, M. A. and Cox, J. W. (2003). A practical approach for large-scale controller performance assessment, diagnosis, and improvement. *Journal of Process Control*, 13(2):155 – 168.

Peng, Y. and Kinnaert, M. (1992). Explicit Solution to the Singular LQ regulation Problem. *IEEE Trans AC*, 37:633–636.

Qin, S. J. and Yu, J. (2007). Recent developments in multivariable controller performance monitoring. *Journal of Process Control*, 17(3):221 – 227. Special Issue ADCHEM 2006 Symposium.

Rahman, A. and Choudhury, M. A. A. S. (2010). New Methods for Detection of Control Loop Interaction to Prioritize Control Loop Maintenance. In *6th International Conference on Electrical and Computer Engineering ICECE 2010*, pages 263 – 266, Dhaka - Bangladesh.

Rahman, A. and Choudhury, M. S. (2011). Detection of control loop interactions and prioritization of control loop maintenance. *Control Engineering Practice*, 19(7):723 – 731.

Rice, R., Jyringi, R., and Cooper, D. (2006). Performance monitoring fundamentals: demystifying performance assessment techniques.

Rice, R. C., Jyringi, R. R., and Cooper, D. J. (2010). Performance Monitoring Fundamentals: Demystifying Performance Assessment Techniques. Technical report, Control Station, Inc.

Rogozinski, M. W., Paplinski, A. P., and Gibbard, M. J. (1987). An Algorithm for the Calculation of a Nilpotent Interactor Matrix for Linear Multivariable Systems. *IEE Transactions on Automatic Control*, 32(3):234 – 237.

Rossi, M., Tangirala, A., Shah, S., and Scali, C. (2006). A data-base measure for interactions in multivariate systems. In *International Symposium on Advanced Control of Chemical Processes ADCHEM 2006*, pages 681 – 686, Gramado - Brasil.

Ruel, M. (2003). The conductor directs this orchestra - Instrument Society of America. *Intech*, 50(11):20–22.

Schäfer, J. and Çinar, A. (2002). Multivariable MPC performance assessment, monitoring and diagnosis. In *Proc. IFAC World Congress*, Barcelona, Spain.

Schafer, J. and Cinar, A. (2004). Multivariable mpc system performance assessment, monitoring, and diagnosis. *Journal of Process Control*, 14(2):113 – 129.

Seborg, D., Edgar, T., and Mellichamp, D. (2004). *Process Dynamics and Control*. John Wiley & Sons.

Skogestad, S. and Postlethwaite, I. (2005). *Multivariable feedback control: analysis and design*. John Wiley.

Stanfelj, N., Marlin, T. E., and MacGregor, J. F. (1993). Monitoring and diagnosing process control performance: the single-loop case. *Industrial & Engineering Chemistry Research*, 32(2):301–314. American Control Conference, 1991.

Swanda, A. and Seborg, D. (1997). Evaluating the performance of PID-type feedback control loops using normalized settling time. In *Proc IFAC ADCHEM*, pages 301–306, Banff - Canada.

Swanda, A. and Seborg, D. (1999). Controller performance assessment based on setpoint response data. In *Proceedings of the 1999 American Control Conference*, volume 6, pages 3863 – 3867, San Diego, CA.

Torres, B., de Carvalho, F., de Oliveira Fonseca, M., and Filho, C. (2006). Performance assessment of control loops - cases studies. In *Proc IFAC ADCHEM,*, Gramado - Brasil.

Tyler, M. and Morari, M. (1995). Performance assessment for unstable and nonminimum-phase systems. In *IFAC Workshop on on-line fault detection and supervision in the chemical process industries*, pages 200–205, Newcastle-upon-Tyne, England.

Tyler, M. L. and Morari, M. (1996). Performance monitoring of control systems using likelihood methods. *Automatica*, 32(8):1145 – 1162.

Wolovich, W. and Falb, P. (1976). Invariants and Canonical Forms under Dynamic Compensation. *SIAM J. Control*, 14:996 – 1008.

Xia, C. and Howell, J. (2005). Isolating multiple sources of plant-wide oscillations via independent component analysis. *Control Engineering Practice*, 13(8):1027 – 1035. Fault Detection, Supervision and Safety of Technical Processes (S afeprocess 2003).

Xia, H., Majecki, P., Ordys, A., and Grimble, M. (2006). Performance assessment of mimo systems based on i/o delay information. *Journal of Process Control*, 16(4):373 – 383.

Yu, J. and Qin, J. (2008a). Statistical MIMO controller performance monitoring. Part I: Data-driven covariance benchmark. *Journal of Process Control*, 18:277 – 296.

Yu, J. and Qin, S. J. (2008b). Statistical {mimo} controller performance monitoring. part ii: performance diagnosis. *Journal of Process Control*, 18(3 - 4):297 – 319. <ce:title>Festschrift honouring Professor Dale Seborg</ce:title>.

Yu, W., Wilson, D. I., and Young, B. R. (2008). Control performance assessment in the presence of valve stiction. In Grigoriadis, K., editor, *The Eleventh IASTED International Conference on Intelligent Systems and Control, ISC 2008*, pages 379–384, Orlando, Florida, USA.

Appendix A

Demonstration of the MIMO Minimum Variance law

The minimum variance control law can easily be obtained if the unitary interactor matrix is known. Consider a multivariate system,

$$Y = G_p U + G_\varepsilon \varepsilon, \tag{A.1}$$

where Y is an output vector, G_p is the proper process transfer function matrix, U is the control signal vector, G_ε is the proper disturbance transfer function matrix and ε is the white noise sequence. The minimum variance control law minimizes the objective function

$$J = E(\tilde{Y}^T \tilde{Y}), \tag{A.2}$$

where J represent the variance of the interactor-filtered output variable $\tilde{Y}$, $\tilde{Y} = q^{-\theta} DY$. Where θ is the order of the interactor matrix and D is the interactor matrix.

The delay free process transfer function matrix $\tilde{G}_p$ is

$$\tilde{G}_p = D G_p \tag{A.3}$$

and

$$G_p = D^{-1} \tilde{G}_p. \tag{A.4}$$

Substituting G_p into equation (A.1),

$$Y = D^{-1} \tilde{G}_p U + G_\varepsilon \varepsilon. \tag{A.5}$$

Multiplying both sides of equation (A.5) by $q^{-\theta}\mathrm{D}$ yields,

$$q^{-\theta}\mathrm{D}\,\mathrm{Y} = q^{-\theta}\mathrm{D}\,\mathrm{D}^{-1}\tilde{\mathrm{G}}_{\mathrm{p}}\mathrm{U} + q^{-\theta}\mathrm{D}\,\mathrm{G}_{\varepsilon}\varepsilon, \tag{A.6}$$

where $q^{-\theta}\mathrm{D}\,\mathrm{Y}$ is equal $\tilde{\mathrm{Y}}$ and $q^{-\theta}\mathrm{D}\,\mathrm{G}_{\varepsilon}$ is equal $\tilde{\mathrm{G}}_{\varepsilon}$,

$$\tilde{\mathrm{Y}} = q^{-\theta}\tilde{\mathrm{G}}_{\mathrm{p}}\mathrm{U} + \tilde{\mathrm{G}}_{\varepsilon}\varepsilon. \tag{A.7}$$

The design of minimum-variance controller requires a perfect system model and a perfect disturbance model and will result in a complete cancellation of the error, except the noise measurement. Satisfy the Diophantine identity

$$\tilde{\mathrm{G}}_{\varepsilon} = q^{-\theta}\mathrm{D}\,\mathrm{G}_{\varepsilon} = \mathrm{F}_0 + \mathrm{F}_1 q^{-1} + \cdots + \mathrm{F}_{\theta-1} q^{-\theta+1} + q^{-\theta}\mathrm{R} = \mathrm{F} + q^{-\theta}\mathrm{R}, \tag{A.8}$$

where F and R are defined by equation (A.8). Substituting equation (A.8) into (A.7) yields

$$\tilde{\mathrm{Y}} = q^{-\theta}\tilde{\mathrm{G}}_{\mathrm{p}}\mathrm{U} + (\mathrm{F} + q^{-\theta}\mathrm{R})\varepsilon, \tag{A.9}$$

where

$$\tilde{\mathrm{Y}} = \tilde{\mathrm{G}}_{\mathrm{p}}\mathrm{U}_{t-\theta} + \mathrm{F}\varepsilon + \mathrm{R}\varepsilon_{t-\theta}. \tag{A.10}$$

The term $\mathrm{F}\varepsilon$ cannot be affected by the control action, because the other terms only change after delay time

$$\sigma_{\tilde{\mathrm{Y}}^2} = \mathrm{E}(\tilde{\mathrm{Y}}^{\mathrm{T}}\tilde{\mathrm{Y}}) \geq \sigma^2_{\mathrm{F}\varepsilon}. \tag{A.11}$$

Therefore,

$$\mathrm{E}(\tilde{\mathrm{Y}}^{\mathrm{T}}\tilde{\mathrm{Y}}) \geq \mathrm{trace}(\sigma^2_{\mathrm{F}\varepsilon}). \tag{A.12}$$

The minimum variance control is achieved when the sum of the two terms of equation (A.10), which start work after the delay time, is set to zero,

$$\tilde{\mathrm{G}}_{\mathrm{p}}\mathrm{U}_{t-\theta} + \mathrm{R}\varepsilon_{t-\theta} = 0. \tag{A.13}$$

This yields

$$\mathrm{U}_{t-\theta} = -\tilde{\mathrm{G}}_{\mathrm{p}}^{-1}\mathrm{R}\varepsilon_{t-\theta}. \tag{A.14}$$

Substituting equation (A.14) into (A.10) yields

$$\tilde{\mathrm{Y}} = \tilde{\mathrm{G}}_{\mathrm{p}}(-\tilde{\mathrm{G}}_{\mathrm{p}}^{-1}\mathrm{R}\varepsilon_{t-\theta}) + \mathrm{F}\varepsilon + \mathrm{R}\varepsilon_{t-\theta} \tag{A.15}$$

and removing brackets

$$\tilde{\mathrm{Y}} = -\mathrm{R}\varepsilon_{t-\theta} + \mathrm{F}\varepsilon + \mathrm{R}\varepsilon_{t-\theta}, \tag{A.16}$$

there are two terms that may be eliminated and

$$\tilde{\mathrm{Y}} = \mathrm{F}\varepsilon. \tag{A.17}$$

This is the same thing that

$$\varepsilon = \mathrm{F}^{-1}\tilde{\mathrm{Y}}. \tag{A.18}$$

Substituting the equation A.18 into A.14 gives the minimum variance control law

$$\mathrm{U}_{t-\theta} = -\tilde{\mathrm{G}}_{\mathrm{p}}^{-1}\mathrm{R}\,\mathrm{F}^{-1}\tilde{\mathrm{Y}}_{t-\theta} \tag{A.19}$$

or

$$\mathrm{U} = -\tilde{\mathrm{G}}_{\mathrm{p}}^{-1}\mathrm{R}\,\mathrm{F}^{-1}\tilde{\mathrm{Y}} \tag{A.20}$$

where $\tilde{\mathrm{Y}} = q^{-\theta}\mathrm{D}\,\mathrm{Y}$ and

$$\mathrm{U} = -\tilde{\mathrm{G}}_{\mathrm{p}}^{-1}\mathrm{R}\,\mathrm{F}^{-1}q^{-\theta}\mathrm{D}\,\mathrm{Y}. \tag{A.21}$$

F and R are defined through the response of the system when there is no measured external disturbance or change on the set-point. The control system only responds to the noise, only operates the transfer function associated to external disturbances. To know the disturbance transfer function, one can determine the model of this transfer function by fitting the data to a model of time series, as for SISO systems. To determine F and R use

$$q^{-\theta}\mathrm{D}\,\mathrm{G}_{\varepsilon} = \mathrm{F} + q^{-\theta}\mathrm{R}, \tag{A.22}$$

where F is $\mathrm{F} = \mathrm{F}_0 + \mathrm{F}_1 q^{-1} + \cdots + \mathrm{F}_{\theta-1}q^{-\theta+1}$. As one can see, this minimum variance control law is only able to minimize variance of the interactor-filtered variable $\tilde{\mathrm{Y}}$.

Appendix B

Rogozinski's algorithm

Considering a matrix of transfer functions $G_p(z)$ with dimension p x m describing a linear time-invariant plant with m-inputs and p-outputs. $G_p(z)$ is a full rank matrix, rank $G_p(z) = \min$ (r,m), and the transfer functions are proper

$$\lim_{z \to +\infty} G_p(z) < \infty. \tag{B.1}$$

The degree of the numerator does not exceed the degree of the denominator. To implement this algorithm is necessary to use a first degree polynomial matrix $U(z)$ that will be called a *row shift polynomial matrix* of order k_i, with dimension m $\times$ m,

$$U(z) = U_0 z + U_1 = \begin{bmatrix} 0 & I_r \\ zI_{k_i} & 0 \end{bmatrix}. \tag{B.2}$$

The matrices U_0 and U_1 are defined by the matrix of coefficients with dimension m $\times$ m. Note that U and U (z) are different

$$U = \begin{bmatrix} U_0 \\ U_1 \end{bmatrix} = \begin{bmatrix} 0_r \\ I_p \\ 0_{k_i} \end{bmatrix}, \tag{B.3}$$

where 0_r and 0_{k_i} are matrices with r and k_i rows of zeros. I_m is the m $\times$ m identity matrix. m is the number of columns of G_p matrix. There is a relationship between these indices, $m = r_i + k_i$. The value of k_i is obtained later during the implementation of algorithm, when does the QR factorization. It is the number of zeros-rows on top of the matrix R.

The transfer matrix $G_p(z)$ can be factored according to the RMFD (2.6). The numerator of

the RMFD is assumed to be a polynomial matrix

$$\mathrm{N}(z) = \mathrm{N}_0 z^{\theta} + \mathrm{N}_1 z^{\theta-1} + \cdots + \mathrm{N}_{\theta}, \tag{B.4}$$

which can be represented by a block matrix of coefficients

$$\mathrm{N} = \begin{bmatrix} \mathrm{N}_0 \\ \mathrm{N}_1 \\ \vdots \\ \mathrm{N}_{\theta} \end{bmatrix}, \tag{B.5}$$

where θ is the degree of the denominator polynomial in the RMFD. This θ is called order of the interactor matrix. In general, N_i is not a full-rank matrix, the leading coefficients N_0, N_1, $\cdots$, N_{θ} may be zero.

Rogozinski presented another definition to help calculate the interactor matrix. If $\mathrm{G_p}$ is a full rank matrix and the transfer functions are proper, there is a unitary interactor that can be calculated as a product of all the matrices S^i,

$$\mathrm{D}(z) = \prod_{i=1}^{j-1} \mathrm{S}^i(z), \tag{B.6}$$

where j is the total number of iterations completed in the Rogozinski's algorithm. Each of these S^i matrices are computed by

$$\mathrm{S}^i(z) = \mathrm{U}^i(z)\mathrm{Q}^i, \tag{B.7}$$

where $\mathrm{U}^i(z)$ is a row shift polynomial matrix of order k_i and Q^i is a non-singular m $\times$ m real matrix (an orthogonal matrix for the factorization of the unitary interactor matrix).

It should be noted that the calculated interactor matrices are not unique. According to Huang (1997), for a given transfer function matrix, two interactor matrices, D_1 and D_2, are "equivalent" if

$$\mathrm{D}_1 = \Gamma \mathrm{D}_2, \tag{B.8}$$

where Γ is an r $\times$ r, unitary, real matrix. The matrix is a unitary matrix, if the product of this matrix with its conjugate transpose is equal to the identity matrix.

After these considerations, it is now possible to present the Rogozinski's algorithm. The algorithm (Rogozinski et al., 1987) consists on:

1. Remove all negative exponents of the transfer function matrix. All mathematical opera-

tors should be used at the same time in all transfer functions matrix

$$\mathrm{G_p}^{\star}(z^{-1}) = \mathrm{G_p}(z). \tag{B.9}$$

2. Set $i = 0$ (number of iteration), $\mathrm{N}^{(0)}(z) = \mathrm{N}(z)$, and $\mathrm{D}^{(0)}(z) = \mathrm{I_m}$ to start the algorithm. $\mathrm{I_p}$ is the identity matrix m $\times$ m.

3. Check whether $r_i = \mathrm{rank}(\mathrm{N}_0^{i-1}) = \min(r,m)$. If this is the case, the algorithm terminates and the unitary interactor matrix is $\mathrm{D}(z) = \mathrm{D}^{i-1}(z)$.

   If $r_i < \min$(r,m), factorize N_0^{i-1} using a matrix factorization method such as Householder orthogonalization QR, Gaussian elimination LDU or singular value decomposition SVD. In this work it is used the QR factorization. The result of this factorization must be

$$\mathrm{N}_0^{(i-1)} = (\mathrm{Q}^{(i)})^{-1} \begin{bmatrix} 0_{k_i} \\ \mathrm{N}_{0\mathrm{D}}^{(i)} \end{bmatrix}, \tag{B.10}$$

   or, equivalently,

$$\mathrm{Q}^{(i)}\mathrm{N}_0^{(i-1)} = \begin{bmatrix} 0_{k_i} \\ \mathrm{N}_{0\mathrm{D}}^{(i)} \end{bmatrix}, \tag{B.11}$$

   where $\mathrm{Q}^{(i)}$ is a m $\times$ m non-singular matrix and k_i is the number of zero lines at the top of the R matrix of the QR factorization ($k_i = \mathrm{m} - r_i$). This value of k_i is needed to determine the matrix U (z).

4. Multiply $\mathrm{N}^{(i-1)}(z)$ by matrix Q^i

$$\bar{\mathrm{N}}(z) = \mathrm{Q}^{(i)}\mathrm{N}^{(i-1)}(z), \tag{B.12}$$

   where $\bar{\mathrm{N}}(z)$ is a new matrix.

5. Multiply $\bar{\mathrm{N}}(z)$ by the row shift polynomial matrix, U (z), of order k_i

$$\mathrm{N}^{(i)}(z) = \mathrm{U}^{(i)}(z)\bar{\mathrm{N}}(z). \tag{B.13}$$

   This multiplications shifts the matrix of coefficients of $\mathrm{N}(z)$, upwards by k_i rows of zeros.

6. Multiply the row shift polynomial matrix, U (z) by the $Q^{(i)}$

$$S^{(i)}(z) = U^{(i)}(z)Q^{(i)}. \tag{B.14}$$

7. Update the matrix D

$$D^{(i)}(z) = S^{(i)}(z)D^{(i-1)}(z), \tag{B.15}$$

where $D^{(i)}$ is the new interactor matrix. This step ends the i^{th} iteration. You must return to step 3, however, we must first determine the N_0 to know the rank of this matrix (see step 3). If the algorithm stops, this matrix, D, is the unitary interactor matrix. The values of the N_0 matrix are the coefficients of the polynomials in the new N matrix (B.13) and that are associated with the exponent equal to the order of the interactor matrix.

Appendix C

Prototype in Octave for the determination of the interactor matrix using the transfer functions

```
% Calculation of interactor matrix
% This program is based in the algorithm presented by Rogozinski,
% Paplinski and Gibbard in 1987(rogozinski-etal87).
% The algorithm will be used to derive the nilpotent interaction matrix.
ny = 2; nu = 2; tol = 1e-10; % number of output variable(ny) manipulated
                             % variable(nu)
% Introduction of transfer functions of the process
% (maximum = 10 x 10 transfer function)
% Each line is a polynomial
% Gp(y1,u1) = A/B; Gpm (1, :) = A; Gpm (2, :) = B;
% Gp(y1,u2) = C/D; Gpm (3, :) = C; Gpm (4, :) = D;
% Transfer function matrix a + b * Z^(-1) + c * Z^(-2) + ...
Gpm   = [ 0     1    0   0   0   0   0   0   0    0   0   0;
          1  -0.4    0   0   0   0   0   0   0    0   0   0;
          0     0  0.5   0   0   0   0   0   0    0   0   0;
          1  -0.1    0   0   0   0   0   0   0    0   0   0;
          0   0.3    0   0   0   0   0   0   0    0   0   0;
          1  -0.1    0   0   0   0   0   0   0    0   0   0;
          0     0    1   0   0   0   0   0   0    0   0   0;
          1  -0.8    0   0   0   0   0   0   0    0   0   0;
          0     0    0   0   0   0   0   0   0    0   0   0;
          0     0    0   0   0   0   0   0   0    0   0   0;
          0     0    0   0   0   0   0   0   0    0   0   0;
```

```
          0     0    0   0   0   0   0   0   0    0   0   0;
          0     0    0   0   0   0   0   0   0    0   0   0;
          0     0    0   0   0   0   0   0   0    0   0   0;
          0     0    0   0   0   0   0   0   0    0   0   0;
          0     0    0   0   0   0   0   0   0    0   0   0;
          0     0    0   0   0   0   0   0   0    0   0   0;
          0     0    0   0   0   0   0   0   0    0   0   0;
          0     0    0   0   0   0   0   0   0    0   0   0;
          0     0    0   0   0   0   0   0   0    0   0   0];
% Verify the input of transfer functions
Gp = cell(ny,nu);
aux=2;
for i = 1 : ny;
    for j = 1 : nu;
       Gp{i,j} = filt(Gpm (aux * (nu * i - nu + j)-1,:),
                 Gpm (aux*(nu * i - nu + j),:),-1);
    endfor
endfor
% To apply this algorithm must remove the negative exponents
dn = zeros(ny*nu,1); dd = zeros(ny*nu,1);
for i = 1 : ny;
    for j = 1 : nu;
       dn(nu * i - nu + j) = degree(Gpm(aux * (nu * i - nu
                             + j)-1,:));
       dd(nu * i - nu + j) = degree(Gpm(aux * (nu * i - nu
                             + j),:));
    endfor
endfor
maxdeg = max ([dn; dd]);
tfm = Gpm(1:2*ny*nu,:);
[n_tf,kt] = size(tfm);
for i=1:maxdeg,
    tfm = [zeros(n_tf,1) tfm];
    for j=1:kt,
         tfm(:,j)    = tfm(:,j+1);
         tfm(:,kt+1) = zeros(n_tf,1);
    endfor
endfor
[n_tft, ktt] = size(tfm);
tfm = tfm(:,1:(ktt-kt+1));
```

```
% Transfer functions without negative exponents
Gpwe = cell(ny,nu);
aux=2;
for i = 1 : ny;
    for j = 1 : nu;
       Gpwe{i,j} = tf(tfm (aux * (nu * i - nu + j)-1,:),
                   tfm (aux*(nu * i - nu + j),:),-1);
    endfor
endfor
% Right Matrix Function Description Gp = N * R^(-1).
% The process transfer function without the time delay must be
% factored into parts, which are denoted as N(z) and R(z)
[n_tft2, ktt2] = size (tfm);
rtfn = cell(ny*nu,1); rtfd = cell(ny*nu,1);
for i = 1 : nu*ny;
    if abs(tfm(2*i, :)) < tol * ones(1,ktt2),
          rtfd{i,1} = [];
    else
          rtfdi  =  roots(tfm(2*i,:))';
          rtfd{i,1}  =  sort(rtfdi);
    endif;
endfor
% lcma calculate the least common multiple (lcm) of roots vector
R = cell(nu,nu);
vrtfd = [];
for i = 1 : ny*nu;
    vrtfd = [vrtfd rtfd{i,1} NA];
endfor
LCM = lcma_nxp(vrtfd,ny,nu);
LCMn = LCM;                          % For calculate N matrix
ILCM = isna(LCM);
nlcm = length(LCM);
lcmNA = [];
for i = 1:nlcm;
    if ILCM(i) == 1;
       lcmNA = [lcmNA i];
    endif
endfor
NR = [];
%Construction of R matrix
```

```
for i = 1: nu;
    for j = 1: nu;
        if i==j;
            R{j,i} = LCM(1:lcmNA(1)-1);
            nR = length(LCM(1:lcmNA(1)-1)); NR = [NR nR];
            LCM = LCM(lcmNA(1)+1:nlcm);
            ILCM = isna(LCM);
            nlcm = length(LCM);
            lcmNA = [];
            for k = 1:nlcm;
                 if ILCM(k) == 1;
                     lcmNA = [lcmNA k];
                 endif
            endfor
        else
            R{i,j} = 0;
        endif
    endfor
endfor
R;
rmax_R  = max(NR);   % degree of No
% Construction of N matrix
N = cell(ny,nu);
vrtfd = [NA vrtfd];
Ivr = isna(vrtfd);
nvr = length(vrtfd);
vrNA = [];
for k = 1:nvr;
    if Ivr(k) == 1;
         vrNA = [vrNA k];
    endif
endfor
H = cell(ny,nu);
for i = 1:ny;
     for j = 1:nu;
         H{i,j} = setdiff(R{j, j}, vrtfd(vrNA(nu * i- nu + j)
                  + 1: vrNA(nu * i - nu + j + 1) - 1));
         N{i,j} = conv(tfm(2*nu*i-2*nu+2*j-1,:),poly(H{i,j}));
     endfor
endfor
```

```
N;
% Determine the initial conditions of algorithm proposed by
% Rogozinski-etal87
D = cell(ny,ny);
for i = 1:ny;
    for j = 1: ny;
         if i == j;
              D{i,j} = 1;
         else
              D{i,j} = 0;
         endif
    endfor
endfor
No = zeros(ny,nu);
for i = 1:ny;
    for j = 1:nu;
        auxno = [];
        auxno = [zeros(1,ny) N{i,j}];
        No(i,j)  = auxno(length(auxno)-rmax_R);
    endfor
endfor
No;
Nbarra = cell(ny,nu);
S = cell(ny,ny);
rankmin = min(ny,nu);
% Begin of algorithm proposed by Rogozinski-etal87
if rank(No) == rankmin,
   D = D;
else
   while rank(No) < rankmin;
        [Q,R] = qr (No);                          % QR Factorization
        if abs(R(1,:)) < tol*ones(1,length(R(1,:))),
           Q = Q; Q = inv(Q);
        else
           CR  = eye(ny);
           CR1 = CR;
           az = 0;              %auxiliar parameter for next loop
           for i=1:ny
                 if abs(R(ny+1-i,:)) < tol*ones(1,length(R(ny
                                    +1-i,:))) && az == 0;
```

```
                CR1(ny+1-i,:) = CR(1,:);
                CR1(1,:) = CR(ny+1-i,:);
                az = 1;
         endif
   endfor
   Q = Q * CR1; Q = inv(Q);
   R = CR1 * R;
endif                                   % Nbarra = Q * N
for i = 1:ny;
    for j = 1:nu;
        aux3 = [0]; naux3 = length(aux3);
        aux4 = [0]; naux4 = length(aux4);
        for k = 1:ny;
            aux3 = Q(i,k) * N{k,j}; naux3 =
                   length(aux3);
            Naux = poladdmod(aux4, naux4 - 1,
                   aux3, naux3 - 1);
            aux4 = Naux; naux4 = length(aux4);
        endfor
        Nbarra{i,j} = Naux;
    endfor
endfor
Nbarra;                     % Obtaining the U matrix
ki = 0; az=0;               % Determination of Ki of the R
for i= 1:ny
    if abs(R(i,:)) < tol*ones(1,length(R(i,:))) && az == 0;
      ki = ki + 1;
    else
      az = 1;
    endif
endfor
ri = ny-ki;                 % Determination of U Matrix
U = [zeros(ri,ny); eye(ny); zeros(ki,ny)];
Uq0 = U(1:ny,:);  Uq1 = U(ny+1:2*ny,:);
Uq = cell(ny,ny);
for i = 1 : ny;
   for j = 1 : ny;
       Uq {i,j} = poladdmod(Uq0(i,j) * [1 0], 1,
                  Uq1(i,j), 0);
   endfor
```

```
endfor                          % N = U * Nbarra
for i = 1:ny;
    for j = 1:nu;
         aux1 = [0]; naux1 = length(aux1);
         aux2 = [0]; naux2 = length(aux2);
         for k = 1:ny;
            aux1 = conv(Uq{i,k},Nbarra{k,j}); naux1
                 = length(aux1);
            Naux = poladdmod(aux1, naux1 - 1, aux2,
                   naux2 - 1);
            aux2 = Naux; naux2 = length(aux2);
         endfor
          N{i,j} = Naux;
    endfor
endfor
N;                              %  S = U * Q
 for i = 1:ny;
    for j = 1:ny;
         aux1 = [0]; naux1 = length(aux1);
         aux2 = [0]; naux2 = length(aux2);
         for k = 1:ny;
            aux1 = conv(Uq{i,k},Q(k,j)) ; naux1
                 = length(aux1);
            Naux = poladdmod(aux1, naux1 - 1, aux2,
                   naux2 - 1);
            aux2 = Naux; naux2 = length(aux2);
         endfor
          S{i,j} = Naux;
    endfor
endfor
S;                              % D = S * D(i-1)
D_1 = D;
for i = 1:ny;
    for j = 1:ny;
         aux1 = [0]; naux1 = length(aux1);
         aux2 = [0]; naux2 = length(aux2);
         for k = 1:ny;
            aux1 = conv(S{i,k}, D_1{k,j}); naux1
                 = length(aux1);
            Naux = poladdmod(aux1, naux1 - 1, aux2,
```

```
                        naux2 - 1);
                    aux2 = Naux; naux2 = length(aux2);
                endfor
                 D{i,j} = Naux;
            endfor
        endfor
        D;                          % Determination New No
        for i = 1:ny;
            for j = 1:nu;
                auxno = [];
                auxno = N{i,j};
                No(i,j)  = auxno(length(auxno)-rmax_R);
            endfor
        endfor
        No;
   endwhile
D
endif
pause
```

Appendix D

ARX Model of the process dynamic obtained with closed loop data

The program of the Appendix G, with the simulated data and considering a number of parameters equal to 5, obtains the following state space model (ARX model).

$$\dot{x} = \mathrm{A}x + \mathrm{B}u \tag{D.1}$$

$$y = \mathrm{C}x + \mathrm{D}u$$

Where,

$$\mathrm{A} = \begin{bmatrix} 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.08 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & -0.10 \\ 0.00 & 1.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.20 \\ 1.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.03 & 0.00 \\ 0.00 & 0.00 & 0.00 & 1.00 & 0.00 & 0.00 & 0.00 & 0.00 & -0.16 & 0.00 \\ 0.00 & 0.00 & 1.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & -0.07 \\ 0.00 & 0.00 & 0.00 & 0.00 & 1.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 1.00 & 0.00 & 0.00 & 0.00 & -0.02 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & -1.00 & 0.00 & 0.51 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 1.00 & 0.00 & 0.79 \end{bmatrix} \tag{D.2}$$

$$\mathrm{B} = \begin{bmatrix} -0.03 & -0.72 \\ 0.00 & -0.03 \\ -0.02 & -0.66 \\ -0.26 & -0.18 \\ -0.03 & -3.01 \\ -0.03 & -0.70 \\ -0.02 & 7.26 \\ 0.21 & -0.86 \\ -1.03 & -0.51 \\ -0.44 & -0.04 \end{bmatrix} \tag{D.3}$$

$$\mathrm{C} = \begin{bmatrix} 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & -1.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & -1.00 \end{bmatrix} \tag{D.4}$$

$$\mathrm{D} = \begin{bmatrix} 0.00 & 0.00 \\ 0.00 & 0.00 \end{bmatrix} \tag{D.5}$$

Appendix E

Determination of the interactor matrix with the Markov parameters

In the multivariate case, the notion of a delay corresponds to the smallest number of impulse response or Markov parameter matrices whose linear combination is nonsingular (det $\neq 0$). This linear combination of impulse response matrices can be expressed in a polynomial matrix form (Huang, 1997)

$$G_p = \sum_{i=0}^{\infty} G_i q^{-i-1}, \tag{E.1}$$

where G_i is the markov parameters matrix associated to each instant of time (q^{-i}). In these matrices, the values also represent the values that would be obtained for an impulse response. When the model of a transfer function or a real process is approximated to a time series model,

$$G_p = \frac{1}{1 - a q^{-1}}. \tag{E.2}$$

This model can directly be transformed into a sum of Markov parameters,

$$G_p = \frac{1}{1 - a q^{-1}}, \qquad \frac{1}{1 - a q^{-1}} = \sum_{i=0}^{\infty} a^i q^{-i}. \tag{E.3}$$

One should know, if a linear combination of the first Markov coefficients or impulse response matrices is singular or not, whether a linear combination of a set of matrices has a full rank. If the set of matrices has a full rank, it means that the process at that instant of time already responds to changes in input.

With this information, Huang (1997) explored this idea for the determination of the order of the interactor matrix. With this order, it is possible to apply the Rogozinski‘s method to

determine the interactor matrix without a process's model.

This path is to use the Markov parameter representation of the transfer function of the process G_p. This path has two options, one is to obtain the Markov parameters directly through a transfer function obtained by a time series model. The other option is to obtain these parameters directly through the data routine of the process (Huang, 1997 and Jelali, 2010). These two options use the algorithm proposed by Rogozinski (Rogozinski et al., 1987) that was already presented.

The interactor matrix is represented by

$$D = D_0 q^{\theta} + D_1 q^{\theta-1} + \cdots + D_{\theta-1} q^{1}, \tag{E.4}$$

where θ is the order of interaction matrix. Applying the definition of the interactor matrix (2.1)

$$\lim_{q^{-1}\to 0} D G_p = K. \tag{E.5}$$

Replacing D and G_p,

$$\lim_{q^{-1}\to 0} [D_0 q^{\theta} + D_1 q^{\theta-1} + \cdots + D_{\theta-1} q^{1}][G_0 q^{-1} + G_1 q^{-2} + \cdots] = K, \tag{E.6}$$

where K is a full rank matrix (rank(K) = min(r,m)). If the set of matrices gives a full rank matrix, K, it means that the process at that instant of time already responds to changes in input. The value of θ is called the **order of interactor matrix**, in this situation is applied the following set of equations:

$$D_0 G_0 = 0; \tag{E.7}$$

$$D_1 G_0 + D_0 G_1 = 0; \tag{E.8}$$

$$\vdots \tag{E.9}$$

$$D_{\theta-1} G_0 + \cdots + D_1 G_{\theta-2} + D_0 G_{\theta-1} = K; \tag{E.10}$$

This may be represented in the following matrix form,

$$[D_{\theta-1},\cdots,D_0]\times\begin{bmatrix} G_0 & 0 & 0 & \cdots & 0 \\ G_1 & G_0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ G_{\theta-2} & G_{\theta-3} & \cdots & \ddots & 0 \\ G_{\theta-1} & G_{\theta-2} & \cdots & \cdots & G_0 \end{bmatrix} = [K,0,\cdots,0], \tag{E.11}$$

simplifying,

$$D^{\circ}G^{\circ} = K^{\circ}, \tag{E.12}$$

where G° is termed as **block-Toeplitz matrix**. D° is the algebraic matrix form of the interactor matrix, while D is the matrix polynomial form. K° is the result of the multiplication. The existence of an exact solution of (E.12) depends on two factors. One factor is that the G° matrix must be invertible, it is invertible if the G° matrix rank is greater or equal to the K matrix rank (E.5) (Huang, 1997 and Jelali, 2010),

$$\text{rank}(G^{\circ}) \geq \text{rank}(K) = \min(r,m). \tag{E.13}$$

If this condition is not satisfied, the block-Toeplitz matrix must be expanded by adding more Markov parameters. The other factor is the order of the interactor matrix θ, the "size" of G°. For determining the order of the interactor matrix, is used the **singular value decomposition, SVD** technique (Huang, 1997 and Jelali, 2010),

$$G^{\circ} = U\Sigma V^{T} = [s_1\, s_2]\begin{bmatrix} v_n & 0 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} d_1^T \\ d_2^T \end{bmatrix}, \tag{E.14}$$

where $[s_1\, s_2]$ and $[d_1\, d_2]^T$ are orthogonal matrices. The columns of s_2 are the columns of U that are associated with singular values equal to zero in the Σ matrix ($s_2^T G^{\circ} = 0$). v_n is a full rank diagonal matrix with singular values of the G°. The rows of d_1^T are the rows of V that are associated with singular values not equal to zero in the Σ matrix. The order of the Interactor should be chosen so that the d_2^T matrix fulfills the following equality,

$$K^{\circ}d_2 = 0. \tag{E.15}$$

This equation can be simplified considering $K^{\circ} = [K, 0, \cdots, 0]$,

$$K^{\circ} d_2 = [K, 0, \cdots, 0] \begin{bmatrix} d_{21} \\ d_{22} \\ \vdots \\ d_{2\theta} \end{bmatrix} = K d_{21}, \tag{E.16}$$

where d_{21} is the upper partition of the matrix d_2, the number of rows of d_{21} is equal to the number of rows of G_p. It can be concluded that equation (E.15) is equal to the following equation

$$K d_{21} = 0. \tag{E.17}$$

If G_p matrix is a square matrix or is an $r \times m$ non-square matrix with $r > m$, the equation E.17 can be simplified to

$$d_{21} = 0. \tag{E.18}$$

If these two conditions are not satisfied, the block-Toeplitz matrix must be expanded by adding more Markov parameters until they are satisfied. If these conditions are satisfied, the order of the interactor matrix θ is defined. A block matrix of the first θ Markov parameters is expressed in a block matrix form as

$$\Lambda = [G_0^T, G_1^T, \cdots, G_\theta^T]^T. \tag{E.19}$$

When this matrix is defined, it remains only to **apply the algorithm proposed by Rogozinski** (Rogozinski et al., 1987) to calculate the interactor matrix. The numerator matrix coefficients of the RMFD of G_p would be replaced by the first θ Markov parameter matrices.

The option to get the interactor matrix using only routine data of process uses the same algorithm described above, the only difference is the Markov parameters obtained from the experimental data (Huang, 1997 and Jelali, 2010). The strategy is to get a time series model that describes the process in open-loop. After getting this model, determines the theoretical response of this model to an impulse and the values obtained for each moment are the Markov parameters.

The model of the open-loop process is required for this method, however, the industry level, the data obtained are in closed loop condition. Although the Markov parameters of the open-loop and closed-loop transfer function matrix are different, their linear combination yields the same interactor matrix (Huang, 1997). The interactor matrix of an open-loop transfer function

can be estimated from the closed-loop data.

The algorithm require only the first θ Markov parameters matrices (or impulse response coefficient matrices). Since the first few Markov parameter matrices contribute to the initial transient response of the process, these parameters characterize the high frequency dynamics of the process. Thus, to estimate this matrix is recommended to use a system identification that use data with high-frequency **dither signal**. The magnitude of the dither signal should be selected such that it has a very weak effect on the process output relative to the existing process disturbances. However, this signal can not be too small because it may be insufficient to correctly identify the model, there is not enough information in the data to identify the dynamics of the process.

This dither signal signal would be applied at the input of the process, the control signal. A random dither signal in the control signal may not be allowed, for such case, this dither signal can be added to the set-point of the process. Instead of the dither signal can be used a simple step changes of the Set-point. However, a low-frequency dither signal may yield a poor estimate of the interactor matrix relatively high-frequency dither signal.

The estimated Markov parameter matrices are not exact due to disturbances, and this makes numerical determination of the rank of the block-Toeplitz matrix G° somewhat arbitrary. To avoid problems in the computation of this method, it is important to omit singular values very close to zero when SVD decomposition is applied. The first step is to define the threshold. One threshold is the largest singular value which is obtained from the Markov parameter matrix at time instant before applying the dither signal. when a singular value is below this threshold, it is necessary to replace this by zero and calculate the matrix of Markov parameters in accordance with this adjustment. The other threshold is a rule of thumb, this rule use a threshold that is equal to a,

$$a = \frac{2}{\sqrt{N_a}}, \tag{E.20}$$

where N_a is the data length. The procedure is the same.

This is a reasonable approximation when a singular value is significantly small. It is recommended to make this change to all matrices with the impulse response values (Markov parameters) before building the block-Toeplitz matrix G° and the block matrix Λ. These two matrices are equivalent relative to those used in the method that uses the Markov parameters of the transfer functions, (E.12) to G° and (E.19) to Λ.

Appendix F

Prototype to determine the interactor matrix with the Markov parameters obtained from time-series models

```
% Calculation of interactor matrix
% This program is based in the algorithm presented by Huang (huang97).
% The algorithm uses the transfer functions (time series format) and
%  transform these into polynomials with coefficients Markov.
% With these coefficients, we construct the right block-Toeplitz matrix
%  and apply the Rogozinski algorithm (rogozinski-etal87).
ny = 2; nu = 2; tol = 1e-10; % number of output variable(ny)
                             %  manipulated variable(nu)
nq = 30;                     % number of markov coefficients
% Introduction of transfer functions of the process
%  (maximum = 10 x 10 transfer function)
% Each line is a polynomial (A or B).
% Gp(y1,u1) = A/B; Gpm (1, :) = A; Gpm (2, :) = B;
% Gp(y1,u2) = C/D; Gpm (3, :) = C; Gpm (4, :) = D;
% Transfer function matrix a + b * Z^(-1) + c * Z^(-2) + ...
Gpm   = [ 0     1     0     0     0     0     0     0     0      0;
          1  -0.4     0     0     0     0     0     0     0      0;
          0     0   0.5     0     0     0     0     0     0      0;
          1  -0.1     0     0     0     0     0     0     0      0;
          0   0.3     0     0     0     0     0     0     0      0;
          1  -0.1     0     0     0     0     0     0     0      0;
          0     0     1     0     0     0     0     0     0      0;
          1  -0.8     0     0     0     0     0     0     0      0;
```

```
          0     0     0     0     0     0     0     0     0     0;
          0     0     0     0     0     0     0     0     0     0;
          0     0     0     0     0     0     0     0     0     0;
          0     0     0     0     0     0     0     0     0     0;
          0     0     0     0     0     0     0     0     0     0;
          0     0     0     0     0     0     0     0     0     0;
          0     0     0     0     0     0     0     0     0     0;
          0     0     0     0     0     0     0     0     0     0];
% Verify the input of transfer functions
Gp = cell(ny,nu);
aux=2;
for i = 1 : ny;
    for j = 1 : nu;
       Gp{i,j} = filt(Gpm (aux * (nu * i - nu + j)-1,:),
        Gpm (aux*(nu * i - nu + j),:),-1);
    endfor
endfor
Gp;
% Determine Markov parameter representation
%        1
%  -------------- = sumatorio{i = 0 to infinito} (a^i * q^(-i))
%  (1 - a*q^(-1))
numG = cell(ny,nu);
denG = cell(ny,nu);
aux  = 2;
for i = 1 : ny;
     for j = 1 : nu;
         denG1 = [];
         numG{i,j} = Gpm (aux * (nu * i - nu + j) - 1,:);
         for k = 1 : nq;
             denG1 = [denG1 (-1 * Gpm (aux * (nu * i - nu + j),
                      2)) ^ (k - 1)];
         endfor
         denG{i,j} = denG1;
     endfor
endfor
% Construction of the matrix with the coefficients of markov
Gpmm = cell(ny,nu);
for i = 1 : ny;
    for j = 1 : nu;
```

```
        Gpmm1 = [];
        Gpmm1 = conv(numG{i,j},denG{i,j});
        Gpmm {i,j} = Gpmm1;
    endfor
endfor
nk = 1;
[G] = toep_matrix (Gpmm, nk, ny, nu);
% Determine order of the interactor matrix by SVD decomposition
rnkGo = rank(G);
[U,E,V] = svd(G);
dE = diag(E);
[nrE, ncE] = size(E);
nzE = 0;
for i = 1:ncE;
     if(abs(dE(i)) < tol)
         nzE = nzE + 1;
     endif
endfor
if nzE == 0;
   nzE = 1;
endif
V21 = V(1:nu,ncE-nzE+1:ncE);
order_interactor = 1;
    while rank(G) <  min(ny,nu) || any(any(abs(V21) >= tol));
        nk = nk+1;
        [G] = toep_matrix (Gpmm, nk, ny, nu);
        [nrG,ncG] = size(G);
        rnkGo = rank(G);
        [U,E,V] = svd(G);
        [nrV, ncV] = size(V);
        dE = diag(E);
        [nrE, ncE] = size(E);
        nzE = 0;
        for i = 1:ncE;
            if(abs(dE(i)) < tol)
                nzE = nzE + 1;
            endif
        endfor
        if nzE == 0;
             nzE = 1;
```

```
        endif
        V21 = V(1:nu,ncE-nzE+1:ncE);
        order_interactor = min([nrG/ny ncG/nu]);
    endwhile
order_interactor = order_interactor
odi = order_interactor;
% Construct a block matrix of the first order_interactor
%  Markov parameters (lambda)
lambda = G(:,1:nu);
[nrla,ncla] = size (lambda);
% Now, apply the recursive algorithm of Rogozinski-etal87
%  to give interactor matrix.
% Determine the initial conditions of algorithm proposed
%  by Rogozinski-etal87
% Construction of N matrix with the block matrix
N = cell(ny,nu);
aux5 = [];
for i = 1: ny;
    for j = 1: nu;
        aux5 = Gpmm{i,j};
        N{i,j} = aux5(1);
    endfor
endfor
No = zeros(ny,nu);
aux6 = 0;
for k = 1:order_interactor;
     for i = 1:ny;
         for j = 1:nu;
             N{i,j} = [N{i,j} lambda(ny*k-ny+i,j)];
             aux6 = Gpmm{i,j};
             No(i,j)= aux6(1);
         endfor
     endfor
endfor
N;
No
% Determine the initial conditions of algorithm proposed
%  by Rogozinski-etal87
D = cell(ny,ny);
for i = 1:ny;
```

```
    for j = 1: ny;
         if i == j;
              D{i,j} = 1;
         else
              D{i,j} = 0;
         endif
    endfor
endfor
Nbarra = cell(ny,nu);
S = cell(ny,ny);
rankmin = min(ny,nu);
% Begin of algorithm proposed by Rogozinski-etal87
pause
```

Appendix G

Prototype to determine the interactor matrix with the Markov parameters obtained from industrial data

```
% Calculation of interactor matrix
% This program is based in the algorithm presented
%  by Huang (huang97).
% The algorithm uses closed-loop data to determine
%  the Markov coefficients.
% With these coefficients, we construct the right block-Toeplitz
%  matrix and apply the Rogozinski algorithm (rogozinski-etal87).
% General parameters
    tol   = 1e-10;
    mnarx = 5;          % order of desirable ARX model
%% load data of process
    f10 = fopen("datamimo2x2.m", "r");
    load datamimo2x2.m; %load datamimo2x2.m;
    fclose(f10)
    [nrY, ncY] = size(Y);
    [nrU, ncU] = size(U);
    [nrr, ncr] = size(r);
    [nrN, ncN] = size(N);
%% Determination of the time series model that translates data
%   from the process(ARX)
    data = iddata(Y,U,dt);
    [sys,x0] = arx(data,mnarx)
%% Impulse response to calculate makov's parameters
```

```
     [yy,tt,xx]=impulse(sys);
     impulse(sys)
tyy = [tt yy(:,:,1) yy(:,:,2)];
    f1 = fopen("inter5.dat","w");
    save inter5.dat tyy;
    fclose(f1);
%% Determine Block- Toeplitz matriz
nbtp = length(yy);
BT  = [];
for i = 1 : nbtp;
    BT1 = [];
    for j = 1:ncU
        BT1 = [BT1 yy(i,:,j)'];
    endfor
    BT = [BT; BT1];
endfor
nbtp2    = length(BT) / ncY;
matrizbt = [];
for i = 1 : nbtp2;
   mbt = BT (ncY * i - ncY +1 : ncY * i, 1:ncU);
   [S, V, D] = svd(mbt);
%   a = 2 / sqrt(length(t))
    a = 0.16;
    dV = diag(V);
    ndV = length(dV);
    for j= 1 : ndV;
        if abs(dV(j)) < a;
             dV(j) = 0;
        endif
    endfor
    V = diag(dV);
    mbt1 = S * V * D';
    matrizbt = [matrizbt; mbt1];
endfor
% Determine order of the interactor matrix
[G] = matrizbt (ncY+1:2*ncY,1:ncU);
rnkGo = rank(G);
[U,E,V] = svd(G);
dE = diag(E);
[nrE, ncE] = size(E);
```

```
nzE = 0;
for i = 1:ncE;
     if(abs(dE(i)) < tol);
         nzE = nzE + 1;
     endif
endfor
if nzE == 0;
   nzE = 1;
endif
V21 = V(1:ncU,ncE-nzE+1:ncE);
nk = 1;
order_interactor = 1;
    while rank(G) <  min(ncY,ncU) || any(any(abs(V21) >= tol));
        nk = nk+1;
        [G] = toep_matrix_data (matrizbt, nk, ncY, ncU);
        [nrG,ncG] = size(G);
        rnkGo = rank(G);
        [U,E,V] = svd(G);
        [nrV, ncV] = size(V);
        dE = diag(E);
        [nrE, ncE] = size(E);
        nzE = 0;
        for i = 1:ncE;
            if(abs(dE(i)) < tol);
                nzE = nzE + 1;
            endif
        endfor
         if nzE == 0;
             nzE = 1;
        endif
        V21 = V(1:ncU,ncE-nzE+1:ncE)
        order_interactor = min([nrG/ncY ncG/ncU]);
    endwhile
order_interactor = order_interactor
odi = order_interactor;
% Construct a block matrix of the first order_interactor
%  Markov parameters (lambda)
lambda = [G(:,1:ncU)];
[nrla,ncla] = size (lambda);
% Now, apply the recursive algorithm of Rogozinski-etal87
```

```
%  to give interactor matrix.
% Determine the initial conditions of algorithm proposed
%  by Rogozinski-etal87
% Construction of N matrix with the block matrix
N = cell(ncY,ncU);
No = zeros(ncY,ncU);
aux6 = 0;
for k = 1:order_interactor+1;
     for i = 1:ncY;
         for j = 1:ncU;
             N{i,j} = [N{i,j} matrizbt(ncY*k-ncY+i,j)];
         endfor
     endfor
endfor
N;
No = matrizbt(1:ncY,1:ncU);
No;
% Determine the initial conditions of algorithm proposed
%  by Rogozinski-etal87
D = cell(ncY,ncY);
for i = 1:ncY;
    for j = 1: ncY;
         if i == j;
              D{i,j} = 1;
         else
              D{i,j} = 0;
         endif
    endfor
endfor
Nbarra = cell(ncY,ncU);
S = cell(ncY,ncY);
rankmin = min(ncY,ncU);
% Begin of algorithm proposed by Rogozinski-etal87
pause
```

Appendix H

MIMO FCOR algorithm

This algorithmo is used to determine the performance of a set of output variables (routine data) of a MIMO control system (Jelali, 2010). It has the following requirements:

- An unitary interactor matrix ($\mathrm{D}(q)$) determined for the system;
- Uncompressed data obtained in a period without set-point changes, normalized to be zero mean;

The method comprises the following steps:

1. Identify a multivariate time series model of the closed-loop data, $\mathrm{G_{p,cl}}$. With this model it is determined the "whitened" sequence $varepsilon(k)$ using

$$\varepsilon(k) = {\mathrm{G_{p,cl}}}^{-1}\mathrm{Y}(k). \tag{H.1}$$

2. Filter the values of the output variables ($Y(k)$) to get the interactor-filtered form $\tilde{Y}(k)$, $\tilde{Y}(k) = q^{-\theta}\mathrm{D}(q)Y(k)$.

3. Consider the decomposition of the interactor matrix,

$$\mathrm{D}(k) = \mathrm{D}_0 q^{\theta} + \cdots + \mathrm{D}_{\theta-1}q. \tag{H.2}$$

As the interaction matrix is unitary, one can make the following operation,

$$\mathrm{D}^{-1}(q) = \mathrm{D}_0^{\mathrm{T}} q^{-\theta} + \cdots + \mathrm{D}_{\theta-1}^{\mathrm{T}} q^{-1}. \tag{H.3}$$

Calculate X

$$\mathrm{X} = [\mathrm{D}_0^{\mathrm{T}}, \mathrm{D}_1^{\mathrm{T}}, \cdots, \mathrm{D}_{\theta-1}^{\mathrm{T}}] \times \begin{bmatrix} \Sigma_{\tilde{Y}\varepsilon}(0)\Sigma_{\varepsilon}^{-1/2} & \Sigma_{\tilde{Y}\varepsilon}(1)\Sigma_{\varepsilon}^{-1/2} & \cdots & \Sigma_{\tilde{Y}\varepsilon}(\theta-1)\Sigma_{\varepsilon}^{-1/2} \\ \Sigma_{\tilde{Y}\varepsilon}(1)\Sigma_{\varepsilon}^{-1/2} & \vdots & \vdots & \\ \vdots & \vdots & & \\ \vdots & \Sigma_{\tilde{Y}\varepsilon}(\theta-1)\Sigma_{\varepsilon}^{-1/2} & & \\ \Sigma_{\tilde{Y}\varepsilon}(\theta-1)\Sigma_{\varepsilon}^{-1/2} & & & \end{bmatrix}, \tag{H.4}$$

where Σ_{ε} is the variance matrix of the "whitened" sequence and $\Sigma_{\tilde{Y}\varepsilon}(i)$ is the covariance between $\tilde{Y}(k)$ and the "whitened"sequence at lag i.

4. The overall process performance index can be calculated as

$$\eta_{\mathrm{cl}} = \frac{\mathrm{trace}(\mathrm{X}\mathrm{X}^{\mathrm{T}})}{\mathrm{trace}(\Sigma_Y)}, \tag{H.5}$$

where Σ_Y is the variance matrix of the output variables.

5. The performance indices for each individual output can be calculated as

$$[\eta_{\mathrm{y1}}, \eta_{\mathrm{y2}}, \cdots, \eta_{\mathrm{yp}}]^{\mathrm{T}} = \mathrm{diag}(\mathrm{X}\mathrm{X}^{\mathrm{T}}\tilde{\Sigma}_Y^{-1}), \tag{H.6}$$

where $\tilde{\Sigma}_Y^{-1} = \mathrm{diag}(\mathrm{diag}(\Sigma_{\mathrm{Y}}))$

Appendix I

Prototype for assess the performance of a MIMO control system with industrial data

```
% Calculation of performance control assessment - system n x p
% This program is based in the algorithm presented
%  by Huang (huang97).
% The algorithm uses closed-loop data to determine
%  the Markov coefficients.
% With these coefficients, we construct the right
%  block-Toeplitz matrix and apply the Rogozinski algorithm
%  (Rogozinski-etal87).
% With the interactor matrix, apply the FCOR method.
% General parameters
    tol = 1e-10;  varN = 1;  %vard1 = 0.05; vard2 = 0.07;
     % Where varN is the variance of white noise of disturbances
     % Where vard1 is the variance of white noise of dither signal
     % Where vard2 is the variance of white noise of dither signal
% Load data of the process
    f10 = fopen("datamimo2x2.m", "r");
    load datamimo2x2.m; %load datamimo2x2.m;
    fclose(f10)
    [rY, nY] = size(Y);
    [rU, nU] = size(U);
    [rr, nr] = size(r);
    [rN, nN] = size(N);
%% Time series models parameters
    mnarx = 10; narmin = 30; narmax = 35;
%% Determination of the interaction matrix
[D] = finter_cldata(tol, mnarx, Y, U, r, N, dt, t)
```

```
% D = cell(2,2);
% D{1,1} = [0 -0.9578 0]; D{1,2} = [0 -0.2873 0];
% D{2,1} = [0 -0.2873 0 0]; D{2,2} = [0 0.9578 0 0];
%% Normalize the process output data (mean centred)
Y = Y - mean(Y);
%% Obtain the whitened sequence E1,E2
[w, A, C, sbc] = arfit2(Y, narmin, narmax, 'zero');
[AR, RC, PE] = mvar(Y,narmax);
E2 = mvfilter([eye(2), -1*AR],eye(2),Y');
E2 = E2';E22 = E2;
E1 = mvfilter([eye(2), -1*A],eye(2),Y');
E1 = E1';E11 = E1;
cov_mvar  = cov(E2)
cov_arfit = cov(E1)
C
%% Determination of interactor order (taudy)
taudy = 0;
aux0 = [];
for i = 1:nY;
    for j = 1:nY;
        aux0 = D{i,j};
        aux0 = rmzerosb(aux0);
        if (length(aux0) - 1) > taudy;
            taudy = length(aux0)-1;
        endif
    endfor
endfor
taudy=taudy
%% Filter Y with q^(-taudy) D   ---> YF
DF = cell(nY,nY);
aux1 = [];
for i = 1:nY;
    for j = 1:nY;
        aux1 = D{i,j};
        aux1 = rmzerosb(aux1);
        aux1 = [zeros(1,taudy + 1-length(aux1)) aux1];
        aux1 = rmzeros(aux1);
        DF{i,j} = aux1;
    endfor
endfor
```

```
DF;
mDF = zeros(nY,nY*(taudy+1));
for k = 1:taudy + 1;
    for i = 1:nY;
        aux2 = [];
        for j = 1:nY;
            aux2 = DF{i,j};
            aux2 = [aux2 zeros(1,taudy+1)];
            mDF(i,j+nY*(k-1)) = aux2(k);
        endfor
    endfor
endfor
mDF;
YF = mvfilter(mDF,[1 0; 0 1], Y');
YF = YF';
YFF = YF;
covYF=cov(YF)
%% Sigma(ye)
SigmaYE0 = cov(YF,E2);
SigmaYE1 = [SigmaYE0];
for i=1:taudy-1
   YF = YF(1+i:length(YF),:);
   E2 = E2(1:length(YF),:);%
   SigmaYE1 = [SigmaYE1; cov(YF,E2)]
endfor
%SigmaE ^(-1/2)
[nrye1,ncye1] = size(SigmaYE1);
SigmaEa = inv(sqrtm(C))%cov(E2)
mzero=zeros(nY,nY);
kappa  = [];
kappa2 = [];
if taudy <= 1;
   kappa = [SigmaYE0*SigmaEa];
else
   for i = 1:taudy
       kappa2 = [SigmaYE1(nY*i-nY+1:nrye1,:); zeros(nY*i-nY,nY)];
       kappa1 = [];
       kappa3 = [];
       for k = 1:taudy;
           kappa3 = kappa2(nY * k - nY + 1:nY*k,:) * SigmaEa;
```

```
            kappa1 = [kappa1; kappa3];
        endfor
        kappa = [kappa kappa1];
    endfor
endif
%Decompose the matrix D
mDT = [];
for k = 1:taudy;
    mDT1 = zeros(nY,nY);
    for i = 1:nY;
        aux3 = [];
        for j = 1:nY;
            aux3 = D{i,j};
            if(length(aux3) - taudy + k -1) <= 0
                  mDT1(i,j) = 0;
            else
                  mDT1(i,j)= aux3(length(aux3)-taudy+k-1);
            endif
        endfor
    endfor
    mDT = [mDT mDT1'];
endfor
mDT;
kappa;
X = mDT*kappa;
XX = X*X'
covY = cov(Y)
indice=trace(XX)/trace(covY)
dcovY = diag(covY);
indice_vector = diag(XX * inv(diag(dcovY)))
pause
```