



Ana Sofia Ramos Brásio

# INDUSTRIAL PROCESSES MONITORING METHODOLOGIES

PhD thesis in Chemical Engineering, supervised by Doctor Natércia Fernandes and Doctor Andrey Romanenko and submitted to the Department of Chemical Engineering, Faculty of Science and Technology, University of Coimbra

2015

• U • C •



UNIVERSIDADE DE COIMBRA



Ana Sofia Ramos Brásio

# INDUSTRIAL PROCESSES MONITORING METHODOLOGIES

PhD thesis in Chemical Engineering, supervised by Doctor Natércia Fernandes  
and Doctor Andrey Romanenko and submitted to the Department of  
Chemical Engineering, Faculty of Science and Technology, University of Coimbra

**Supervisors:**

Doctor Natércia Fernandes

Doctor Andrey Romanenko

**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

2015



UNIVERSITY OF COIMBRA





*"The best way to predict your future is to create it."*

*Alan Kay*



# Resumo

Na indústria de processos, faz-se permanentemente um esforço considerável para conseguir alcançar objectivos económicos e, simultaneamente, garantir o cumprimento das especificações de qualidade, das restrições operacionais e da regulamentação de segurança e ambiental. O sistema de controlo industrial, composto por centenas a milhares de anéis de controlo, tem à sua responsabilidade a satisfação destes objetivos independentemente do modo de operação da fábrica. Não obstante, fatores como alterações na composição das matérias-primas e envelhecimento, desgaste e incrustações nos equipamentos podem conduzir à degradação do desempenho dos anéis de controlo mesmo que o seu projeto e comissionamento tenham sido feitos apropriadamente. Portanto, os anéis de controlo industriais devem ser permanentemente monitorizados e mantidos com recurso a técnicas e ferramentas automáticas capazes de identificar a degradação do seu desempenho e as causas-raiz para correção e/ou manutenção apropriadas. A presente dissertação segue duas direções fundamentais: a monitorização e a melhoria de desempenho dos anéis de controlo industriais.

Uma vez que as técnicas em questão baseiam-se na identificação de sistemas, esta área mereceu especial atenção, com ênfase em duas aplicações em sistemas típicos no âmbito da Engenharia Química onde são explorados aspectos de implementação. Estas aplicações consideraram um modelo SISO de um permutador de calor industrial e um modelo MIMO de um CSTR simulado. Os casos de estudo apresentados cobrem a seleção da estrutura dos modelos, a estimativa de parâmetros e a abordagem para ultrapassar dificuldades de implementação, num compromisso entre a capacidade de previsão e a complexidade dos modelos.

A monitorização do desempenho dos anéis de controlo foi estudada com ênfase no fenómeno de stiction, a falha mais comum em válvulas de controlo. Foi desenvolvida uma taxonomia das abordagens existentes, cobrindo mais de 150

---

publicações científicas, de modelação, detecção / quantificação e compensação de stiction. Como a maioria dos métodos de diagnóstico de stiction funcionam bem apenas quando as oscilações induzidas pela falha são corretamente detetadas, propõe-se um método de deteção e caracterização de oscilações que revela um desempenho significativamente melhor do que as técnicas descritas na literatura. A técnica, automática e computacionalmente mais leve, diagnosticou corretamente conjuntos de dados exigentes contendo ruído e sobreposição de oscilações.

São também propostas nesta tese duas novas abordagens para o diagnóstico de stiction. A primeira é baseada em otimização numérica e em modelos do processo e de stiction. Dado que a modelação de stiction é caracterizada por descon- tinuidades, é sugerida uma estratégia de suavização para permitir a utilização de técnicas de otimização contínua. A segunda abordagem estende o trabalho de Yamashita (2006a) permitindo lidar com processos integradores, como anéis de controlo de nível. O método proposto transforma os dados de modo a obter uma relação direta entre a variável controlada e a posição da válvula. O método foi aplicado com sucesso a dados simulados e industriais. Apesar de ficar ofus- cado pela presença de ruído, a correta deteção de stiction é possível através do uso de um filtro.

No que respeita à melhoria do desempenho dos anéis de controlo, investiga- ram-se duas soluções nesta tese: a sintonização de controladores PID e a utiliza- ção de sensores inferenciais. Tipicamente, stiction não é considerado explicita- mente na sintonização dos controladores e os métodos disponíveis geralmente baseiam-se na adição de sinais específicos à variável manipulada ou de blocos es- peciais ao algoritmo PID base. Contudo, estas tarefas não são triviais. O método alternativo proposto requer apenas a ressintonização do controlador. Os parâ- metros de sintonização são determinados a partir de um problema de otimização que penaliza quer o desvio entre as variáveis controlada e de referência quer o movimento da válvula. Novos critérios e restrições de desempenho podem ser adicionados para definir a resposta desejada em anel fechado. O método foi apli- cado com sucesso em casos contendo uma válvula saudável e com stiction. O comportamento em anel fechado foi significativamente melhorado em ambos os casos reduzindo o movimento de controlo e as oscilações na variável controlada.

Finalmente, a utilização de sensores inferenciais foi abordada como uma forma



---

de gerar informação que não está prontamente disponível a partir da instrumentação instalada ou de medições laboratoriais. Esta tese apresenta uma aplicação considerando o processo de concentração de glicerina. Desenvolve-se um sensor inferencial para prever a qualidade do produto por forma a minimizar atrasos das medições e permitir ações de controlo mais atempadas. Procede-se, ainda, a um estudo comparativo da capacidade de previsão dos vários modelos utilizados usando dados de treino e de validação.

**Palavras-chave:** *monitorização e melhoria do desempenho de processos industriais, deteção e diagnóstico de falhas, stiction em válvulas de controlo, sintonização de controladores PID, tecnologia de sensores inferenciais, aplicações*



# Abstract

In process industries, significant efforts are continuously made to achieve economic objectives while complying with product quality specifications, constraints of the operation, safety and environmental regulations. The industrial control system composed by hundreds to thousands of control loops aims to satisfy these objectives in all modes of operation of the plant. Nevertheless, some factors such as raw materials composition changes, aging, wear, fouling, and other modifications in the equipment may lead to the degradation of the control loop performance even if the initial design and commissioning was properly carried out. Therefore, industrial control loops should be continuously monitored and maintained with automatic techniques and tools capable of identifying control loop performance degradation and their root causes for proper correction and/or equipment maintenance. The present dissertation follows two fundamental directions: performance monitoring and performance improvement of industrial control loops.

Since the concerned techniques are based on system identification, their implementation aspects were addressed in two chemical engineering applications. The applications considered a SISO model of an industrial heat exchanger and a MIMO model of a simulated CSTR. The case studies present the model structure selection, the parameter estimation, and the approach to overcome some implementation difficulties, taking into account a compromise between the prediction capacity and the model complexity.

The performance monitoring of industrial control loops was studied with a focus on the stiction phenomenon, the most common control valve fault. A taxonomy of existing approaches for the modeling, detection /quantification, and compensation of stiction was developed covering more than 150 publications. Most stiction diagnosis methods work well only when stiction induced oscilla-

---

tions are clearly detected, making the latter a critical issue. In this context, a new method of detection and characterization of multiple oscillations was proposed that has a significantly better performance over existing approaches reported in the literature. This automatic and computationally light approach successfully diagnosed challenging datasets containing noise and multiple frequency oscillations.

Two new approaches for stiction diagnosis were proposed in this thesis. The first is based on the numerical optimization and on process and stiction models. Because stiction modeling is characterized by discontinuities, a smoothing approach was applied to enable the use of continuous optimization techniques. The second extends the work of Yamashita (2006a) to handle integrating processes, such as level control loops. The proposed method transforms the dataset in order to obtain a direct relation between the controlled variable and the control valve position. The method was applied with success to simulated and industrial datasets. Although the stiction phenomenon gets obfuscated by the noise, correct stiction detection is possible using proper data filtering.

Regarding control loop performance improvements, two solutions were investigated in this thesis: the PID controller tuning and the use of soft sensor technology. Typically, stiction is not considered explicitly in controller tuning and the available methods usually rely on specially crafted signals added to the manipulated variable or on the addition of a special block to the nominal PID algorithm. However, these are not trivial tasks. The proposed alternative method requires the retuning of the controller only. It determines the tuning parameters from the solution of an optimization problem whose objective function penalizes the deviation of the controlled variable from the setpoint and the valve movement. Besides, additional performance criteria and constraints may be added in order to define the desired closed-loop response. The approach was successfully applied to two cases with a healthy and a sticky control valve. Furthermore, the closed-loop behaviour was significantly improved in both cases reducing the control moves and oscillations in the controlled variable.

Finally, the use of soft sensor technology was also addressed in the present thesis as a way to generate new information that is not readily available from on-line instrumentation or laboratory measurements. This thesis provides a case



---

study considering the glycerine concentration process. A soft sensor is developed for the prediction of the final product quality to minimize measurement delays and enable quick control actions. The prediction capability of several modeling techniques is compared using training and validation datasets.

**Keywords:** *industrial process performance monitoring and improvement, fault detection and diagnosis, control valve stiction, PID controller tuning, soft sensor technology, applications*



# Acknowledgements

The conclusion of this thesis is undoubtedly an important milestone in my life and the result of a path marked by moments and people who have accompanied, helped, and guided me over these years.

In first place, I would like to start by thanking to my advisors Dr. Natércia Fernandes and Dr. Andrey Romanenko for their trust and the amazing opportunity to enroll the PhD program at University of Coimbra and for coming along with me in this journey. Their excellent supervision, creative ideas, constructive feedback, and friendship were essential in the development of this work. About Dr. Natércia Fernandes, she is a symbol of scientific rigor and perfection I always admire and I will always search after. Dr. Andrey Romanenko has opened my mind to the practical knowledge and has been to me an essential connection between university and industry. It is a privilege to work with people who never sees the work as done and finds always room for improvement.

I would like to thank to Dr. Lino Santos for awakening in me a special interest not only in the fundamental fields of modeling, simulation, and control of chemical processes but also in Linux operating system and the  $\text{\LaTeX}$  typesetting system that are intrinsic tools in my life.

This work was developed under the project NAMPI, reference 2012/023007, in consortium between Ciengis, SA and UC with financial support from QREN via Mais Centro operational regional program and European Union via FEDER framework program and allowed me to proceed with this thesis.

I am also deeply thankful to my lovely parents Elizabete and Carlos for their amazing support and care. Moreover, I wish to thank to my sisters Patrícia and

---

Carolina and to Marco for their essential support that each, in his own way, gave me. Their love and understanding helped me throughout this journey.

To them I specially dedicate this work.



*To Elizabete and Carlos.*



# Contents

<b>Resumo</b>	<b>v</b>
<b>Abstract</b>	<b>ix</b>
<b>Acknowledgements</b>	<b>xiii</b>
<b>List of Tables</b>	<b>xxi</b>
<b>List of Figures</b>	<b>xxiii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Scope and motivation . . . . .	1
1.2 Goals . . . . .	4
1.3 Overview . . . . .	5
<b>2 System Identification</b>	<b>9</b>
2.1 Definition . . . . .	9
2.2 State-of-the-art . . . . .	10
2.2.1 Closed-loop identification . . . . .	10
2.2.2 Fundamental steps . . . . .	12
2.2.3 Selection of the model structure . . . . .	13
2.2.4 Parameter estimation . . . . .	26
2.2.5 Model validation . . . . .	47
2.3 Using first- and second-order state-space models for system identification . . . . .	51
2.3.1 Parameter estimation based on numerical optimization . . . . .	51
2.3.2 Discussion of results . . . . .	53

2.4	Hybrid modeling of a biodiesel decanter . . . . .	59
2.4.1	Liquid-liquid equilibrium . . . . .	61
2.4.2	Dynamic mathematical model of a decanter . . . . .	62
2.4.3	Neural network development and performance . . . . .	65
2.4.4	Discussion of results . . . . .	73
<b>3</b>	<b>Control Loop Performance Assessment</b>	<b>79</b>
3.1	Importance and characteristics . . . . .	79
3.2	State of the art . . . . .	82
3.2.1	Poor performance detection . . . . .	82
3.2.2	Poor performance diagnosis . . . . .	100
3.2.3	Performance improvement . . . . .	114
3.2.4	Stiction modeling, detection/quantification and compensation . . . . .	115
3.3	Detection and characterization of oscillating disturbances . . . . .	159
3.3.1	Auto-correlation function . . . . .	160
3.3.2	Proposed approach . . . . .	162
3.3.3	Discussion of results . . . . .	167
3.4	Stiction detection and quantification through numerical optimization	176
3.4.1	Proposed approach . . . . .	176
3.4.2	Smoothing of discontinuous models . . . . .	178
3.4.3	Development for a system containing a sticky valve . . . . .	179
3.5	Detection of stiction in level control loops . . . . .	185
3.5.1	Yamashita's method . . . . .	186
3.5.2	Proposed approach . . . . .	188
3.5.3	Application to a simulated system . . . . .	189
3.5.4	Influence of noise in the detection . . . . .	193
<b>4</b>	<b>PID controllers tuning</b>	<b>199</b>
4.1	The PID controller . . . . .	199
4.2	State-of-the-art . . . . .	203
4.2.1	Tuning methods . . . . .	203
4.2.2	Automatic tuning . . . . .	217
4.3	Compensation of control valve faults by PID controller tuning . . . . .	217

4.3.1	Proposed approach . . . . .	218
4.3.2	Discussion of results . . . . .	222
<b>5</b>	<b>Soft sensing technology</b>	<b>229</b>
5.1	Importance and definition . . . . .	229
5.2	State-of-the-art . . . . .	231
5.2.1	Development methodology . . . . .	231
5.2.2	Data pre-processing . . . . .	234
5.2.3	Techniques for soft sensing . . . . .	249
5.2.4	Special techniques for soft sensing . . . . .	249
5.2.5	Detection and handling of concept drift . . . . .	259
5.3	Glycerine evaporator application . . . . .	266
5.3.1	Glycerine concentration process . . . . .	267
5.3.2	Soft sensor development . . . . .	268
5.3.3	Effect of model structure on prediction . . . . .	273
<b>6</b>	<b>Conclusions and future work</b>	<b>279</b>
6.1	Main contributions . . . . .	279
6.1.1	System Identification . . . . .	279
6.1.2	Control Loop Performance Assessment . . . . .	280
6.1.3	PID controller tuning . . . . .	281
6.1.4	Soft sensor technology . . . . .	282
6.1.5	Publications . . . . .	282
6.2	Future work . . . . .	284
6.2.1	System Identification . . . . .	284
6.2.2	Control Loop Performance Assessment . . . . .	285
6.2.3	PID controller tuning . . . . .	286
6.2.4	Soft sensor technology . . . . .	286
	<b>Bibliography</b>	<b>289</b>
	<b>Appendices</b>	<b>341</b>
	<b>A CSTR model</b>	<b>A.1</b>
	<b>B Benchmarks modifications and extensions</b>	<b>B.1</b>

<b>C</b>	<b>Missing data</b>	<b>C.1</b>
C.1	Methods for assessing MCAR mechanisms . . . . .	C.1
C.2	Methods for handling missing data . . . . .	C.3
C.3	Application to a small dataset . . . . .	C.6

# List of Tables

2.1	Identification results for the SISO system using both FO and SO models. . . . .	54
2.2	Identification results for the MIMO system using FO and SO models.	57
2.3	Molar volume of ester, methanol and glycerol. . . . .	64
2.4	Average and standard deviation for the normalization of the input data. . . . .	67
3.1	Overview of the performance assessment methods. . . . .	84
3.2	Advanced performance benchmark overview. . . . .	98
3.3	Description of the most common non-linearities of the control valves.	113
3.4	Performance assessment software. . . . .	157
3.5	Methods analysed. . . . .	167
3.6	Results of data analysis by different methods. . . . .	169
3.7	Advantages and disadvantages exhibited by oscillation detection and/or characterization methods . . . . .	175
3.8	MSE for different values of $r$ . . . . .	183
3.9	System identification results. . . . .	185
3.10	Stiction detection results for free-noise closed-loop data of the level control. . . . .	191
3.11	Influence of noise in stiction detection by the three compared methods. . . . .	194
3.12	Stiction detection results for level control industrial data by the three compared methods. . . . .	197
4.1	Step response Ziegler–Nichols tuning method formulae. . . . .	206
4.2	Cohen–Coon method formulae. . . . .	206

## List of Tables

---

4.3	Performance criteria based tuning method constants. . . . .	208
4.4	IMC tuning method formulae. . . . .	209
4.5	Step response MIGO tuning method formulae for self-regulating and integrating processes. . . . .	209
4.6	Frequency response Ziegler–Nichols tuning method formulae. . . . .	210
4.7	Frequency response MIGO tuning method formulae. . . . .	211
4.8	Fault detection results. . . . .	224
4.9	Process and fault+process modeling results. . . . .	224
4.10	PID controller tuning results. . . . .	227
5.1	Performance measures. . . . .	274
A.1	CSTR model parameters. . . . .	A.2
B.1	Modifications and extensions of the MVC benchmark. . . . .	B.1
B.2	Modifications and extensions of the user-specified performance in- dices. . . . .	B.3
B.3	Modifications and extensions of the model-based performance in- dices. . . . .	B.4
C.1	Dataset. . . . .	C.7
C.2	Results of the univariate $t$ -test based approach. . . . .	C.8
C.3	Results of the multivariate Little MCAR test. . . . .	C.8
C.4	Results of the maximum likelihood method. . . . .	C.9
C.5	Imputed dataset $s$ using data augmentation.. . . .	C.10



# List of Figures

1.1	Block diagram showing the components and the signal flow of a control loop. . . . .	2
1.2	Process improvement based on control loop performance monitoring. . . . .	4
1.3	Thesis organization and overview. . . . .	6
2.1	Block diagrams for open- and closed-loop systems. . . . .	10
2.2	Amplitude modulated pseudo-random binary signal. . . . .	12
2.3	Classification of models by their linearity. . . . .	15
2.4	AR model structure. . . . .	18
2.5	ARX model structure. . . . .	19
2.6	OE model structure. . . . .	19
2.7	Box-Jenkins model structure. . . . .	20
2.8	Block-oriented non-linear models. . . . .	23
2.9	Classification of time delay estimation methods. . . . .	46
2.10	SISO system identification using an FO model. . . . .	53
2.11	SISO system identification using an SO model. . . . .	54
2.12	MIMO system identification using an FO model. . . . .	56
2.13	Difficulties in identifying the MIMO system via an SO model. . . . .	58
2.14	MIMO system identification using an SO model. . . . .	59
2.15	Simplified representation of the batch biodiesel production process. . . . .	60
2.16	Flowchart of the <i>flash</i> method to determine the liquid-liquid equilibrium. . . . .	62
2.17	Schematic representation of the decanter. . . . .	63

## List of Figures

---

2.18	Simulation of the transesterification reaction of sunflower oil with model. . . . .	66
2.19	Neural network used to substitute the <i>flash</i> calculation. . . . .	68
2.20	Evolution of the average square error while applying the Levenberg-Marquardt algorithm. . . . .	69
2.21	Prediction of the split fractions through the neural network model. . . . .	70
2.22	Absolute error between the predictions of the neural network and the of the <i>flash</i> calculation. . . . .	71
2.23	Prediction of the split fractions as function of temperature for two different mixtures A and B. . . . .	72
2.24	Variables profiles under the start-up of the decanter operation. . . . .	74
2.25	Variables profiles under disturbances to the operation. . . . .	75
3.1	Block diagram of a single loop system. . . . .	80
3.2	Control performance assessment stages. . . . .	82
3.3	Regulatory control performance assessment methods. . . . .	83
3.4	Deterministic metrics for performance evaluation in the feedback loop defined by $H(s) = G(s)/(1 + G(s))$ with $G(s) = 1/(s^3 + 3s + 2s)$ . . . . .	86
3.5	Robustness metrics for performance evaluation in the feedback loop. . . . .	86
3.6	Control chart representation. . . . .	88
3.7	Control charts used in statistical process control. . . . .	88
3.8	CUSUM control chart demonstrating an out of control process above the upper arm through the V-mask procedure. . . . .	91
3.9	Poor performance detection and diagnosis. . . . .	101
3.10	Methods for oscillating disturbances diagnosis. . . . .	102
3.11	Variables meaning of Miao and Seborg (1999) method. . . . .	105
3.12	Manual diagnosis of oscillating disturbances. . . . .	108
3.13	Phase plot showing the typical behavior of a sticky valve. . . . .	118
3.14	Distinction of limit cycles due to stiction and to other causes. . . . .	118
3.15	Compilation of the modeling approaches. . . . .	120
3.16	Friction force. . . . .	121
3.17	Choudhury Model. . . . .	130
3.18	Kano Model. . . . .	131
3.19	He Model. . . . .	132

3.20	Chen Model. . . . .	133
3.21	Karthiga Model. . . . .	134
3.22	Compilation of the detection /quantification methods. . . . .	135
3.23	Horch method application on signals $u$ (dashed line) and $y$ (solid line). . . . .	136
3.24	Probability density functions characteristic of the stiction and non-stiction cases. . . . .	137
3.25	Detection of stiction through the fitting of an ellipse. . . . .	138
3.26	Shape of the control error for valve stiction and aggressive control. . . . .	138
3.27	Compilation of the compensation methods. . . . .	149
3.28	Diagram illustrating the knocker used in a feedback loop. . . . .	151
3.29	Signal sent by the knocker compensator. . . . .	151
3.30	Signal sent by the two moves compensator. . . . .	153
3.31	Family tree of models and detection /quantification and compensation methods of stiction. . . . .	158
3.32	Process block diagram. . . . .	162
3.33	Proposed method flowchart. . . . .	164
3.34	Simulated datasets. . . . .	168
3.35	Industrial datasets. . . . .	169
3.36	Application of method A to data D4. . . . .	171
3.37	Graphical representation of the application of methods B and C to dataset D2. . . . .	171
3.38	Application of method D to dataset D5. . . . .	173
3.39	Auto-correlation function used by method E for analysis of dataset D5. . . . .	174
3.40	Industrial control loop representing the Hammerstein Model, where $y_{sp}$ is the variable setpoint, $u$ is the controller output, $x$ is the real valve position, and $y$ is the controlled variable. . . . .	176
3.41	Enhanced Chen Model flow diagram. . . . .	181
3.42	Comparison between the Hammerstein Model using the original Chen Model and its smoothed version for different values of $r$ . . . . .	183
3.43	System identification. . . . .	184

## List of Figures

---

3.44	Industrial control loop with stiction, where $y_{sp}$ is the variable set-point, $u$ is the controller output, $x$ is the real valve position, and $y$ is the controlled variable. . . . .	185
3.45	Typical patterns of a sticky valve. . . . .	186
3.46	Symbols used to represent a signal (top) and typical qualitative shapes found in sticky valves (bottom). . . . .	187
3.47	Real valve position $x$ , controlled variable $y$ and transformation function $f(y)$ applied to the level for no stiction and stiction cases. . . . .	189
3.48	Open-loop response of a level control loop obtained by simulation using the Choudhury Model. . . . .	191
3.49	Closed-loop response of a level control loop obtained by simulation using the Choudhury Model. . . . .	192
3.50	Influence of noise. . . . .	195
3.51	Stiction detection results for industrial data. . . . .	198
4.1	Block diagram highlighting a PID controller inside the feedback control loop. . . . .	201
4.2	Classification of single-loop tuning methods with some examples of existing methods. . . . .	204
4.3	Step response Ziegler-Nichols method. . . . .	205
4.4	Zareba method flowchart. . . . .	212
4.5	Classification of multi-loop tuning methods with some examples of existing methods. . . . .	213
4.6	Industrial control loop. . . . .	218
4.7	Proposed approach flowchart. . . . .	219
4.8	Hammerstein Model. . . . .	222
4.9	Flow control loop. . . . .	223
4.10	Dataset 1 results. . . . .	225
4.11	Dataset 2 results. . . . .	226
5.1	Methodology for the soft sensors development. . . . .	232
5.2	Problems present in industrial data. . . . .	235
5.3	Methods for handling collinearity in data. . . . .	236
5.4	Outliers detection. . . . .	239

5.5	Influence of the missing mechanisms on the probability distribution.	242
5.6	Methods for handling missing data. . . . .	243
5.7	Industrial datasets with different degrees of compression identified at the right $y$ -axis. . . . .	245
5.8	Effect of compression on data-driven analysis for different datasets.	246
5.9	Methods for soft sensing. . . . .	250
5.10	Representation of the elbow in the cost function. . . . .	256
5.11	Learnt topology gating artificial neural network. . . . .	258
5.12	Concept drift patterns that may occur over time. . . . .	259
5.13	Concept drift handling methods. . . . .	262
5.14	Diagram of the glycerine concentration process. . . . .	269
5.15	Soft sensor input and output variables. . . . .	270
5.16	Selected data. . . . .	270
5.17	ANN based models representation. . . . .	272
5.18	Results of the model training step. . . . .	273
5.19	Results of the model validation step. . . . .	275
5.20	Effect of PLS model components number in the glycerine composition prediction. . . . .	276
5.21	Effect of FANN model hidden layer neurons number in the glycerine composition prediction. . . . .	276
5.22	Effect of LRANN model hidden layer neurons number in the glycerine composition prediction. . . . .	277
5.23	Effect of LRANN model time delay in the glycerine composition prediction. . . . .	277
C.1	Time-series plot for the simulated $A$ and $C$ means. . . . .	C.11



# Chapter 1

## Introduction

This chapter presents the scope, the motivations, and the goals of the research work and outlines the structure of the thesis.

### 1.1 Scope and motivation

Industrial processes contain multiple variables such as temperatures, compositions, flow rates, levels, and pressures that have multivariable and nonlinear dependencies and are subject to disturbances. Thus, it is not trivial to operate an industrial process so that the economic objectives are achieved while complying with product quality specifications, constraints of the operation, safety and environmental regulations in the presence of various variability factors.

The role of an industrial control system is to satisfy the described objectives during the conversion of the raw materials into the final products. Industrial control systems may be composed by hundreds to thousands of control loops. A control loop is an interconnection of several components forming a configuration that has the goal of influencing the behaviour of process variables in a desired way. The block diagram in Figure 1.1 shows those components: the sensor, the controller (usually of proportional-integral-derivative type), the actuator (usually a control valve), and the process. Being the process the central component because it contains the characteristic variable that one wants to control (the controlled variable), the remaining components are placed around. Hence, the flow of the signal is the following: (1) the current controlled variable value is mea-

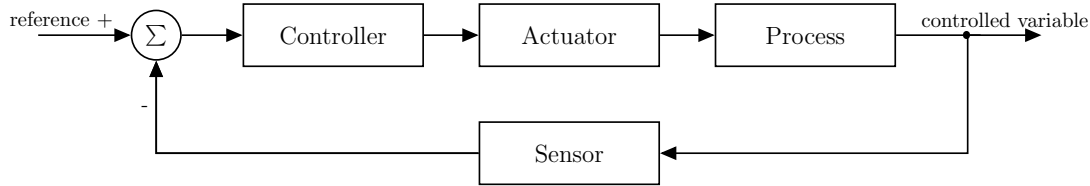


Figure 1.1: Block diagram showing the components and the signal flow of a control loop.

sured by the sensor; (2) the control error is calculated as a difference between the reference and measured signals; (3) the controller uses this error to calculate the control order to the actuator; (4) the control order is implemented by the actuator but, for several reasons, the actual position of the actuator may differ from the order; (5) the process reacts to the actuator action generating a new value for the controlled variable. These steps are repeated continuously.

In an initial phase, the industrial control system is designed, tuned and implemented in order to guarantee a desirable control loop performance.

Even if these tasks are properly carried out, raw materials composition changes, aging, wear, fouling, and some other modifications in the equipment may lead to the degradation of the control system performance after some time in operation. Therefore, control systems must be monitored and maintained. Traditionally, the monitoring task is performed by the control and the maintenance engineers. However, the numerous tasks that engineers are responsible for and the continuous demand for better product quality, higher productivity, and compliance require that the task of monitoring be accomplished routinely and automatically. This context highlights the importance of techniques and tools capable of identifying control loop problems and their root causes for proper correction and/or equipment maintenance.

The most common control loop problems present in industry are intimately related to the control loop components. Regarding the controller, its initial design and commissioning is usually performed in order to respond quickly and appropriately to process load disturbances and setpoint changes. In the common practice, a standardized design is used in all the controllers and only when the poor performance is really noticeable, its parametrization is customized. Besides, the control loop performance may decrease due to changes in process operating



conditions.

The actuator, also called the final control element, is a critical component. In the process industries, the most common final control element is the control valve. Its functioning may deteriorate due to malfunctions such as stiction, backlash, and deadband phenomena. In fact, stiction is one of the long-standing problems causing persistent oscillations and undermining economic performance of the production assets. Although the definitive solution for a faulty valve is to perform maintenance work on the equipment, it is seldom possible to service a critical valve between turnarounds because of operation and safety considerations. Although it is rarely applied in practice, valve fault compensation may mitigate the performance loss until the maintenance is possible. Besides, it may contribute to the extension of control valves life time and to the reduction of maintenance costs.

Another fact that does not allow to maintain the maximum economic performance is the lack of reliable analyzers that measure key process variables beyond simple sensors. This may happen either for economical reasons because of the equipment or maintenance cost or because the measurement device or principle is not available on the market. Besides, laboratory analysis approaches of determination of key quality variables pose a problem for closed-loop control and monitoring because of the typical large delays between the sampling instant and the moment the lab result is ready. The inference of these variables via soft sensing may potentially provide additional process knowledge at a moderate investment.

All these aspects are resumed in Figure 1.2 where a control loop is monitored by using its data stored in a database and improvements are available to apply in the control loop components performing with low performance. It is noteworthy that although advanced process control is out of this thesis scope, it may be used to improve the process performance by changing automatically the operational conditions imposed by the reference value (and that are usually defined manually by the control engineer).

In this context, this thesis in the area of control loop performance monitoring and improvement aims to contribute to the optimization of the economic performance of industrial plants worldwide.

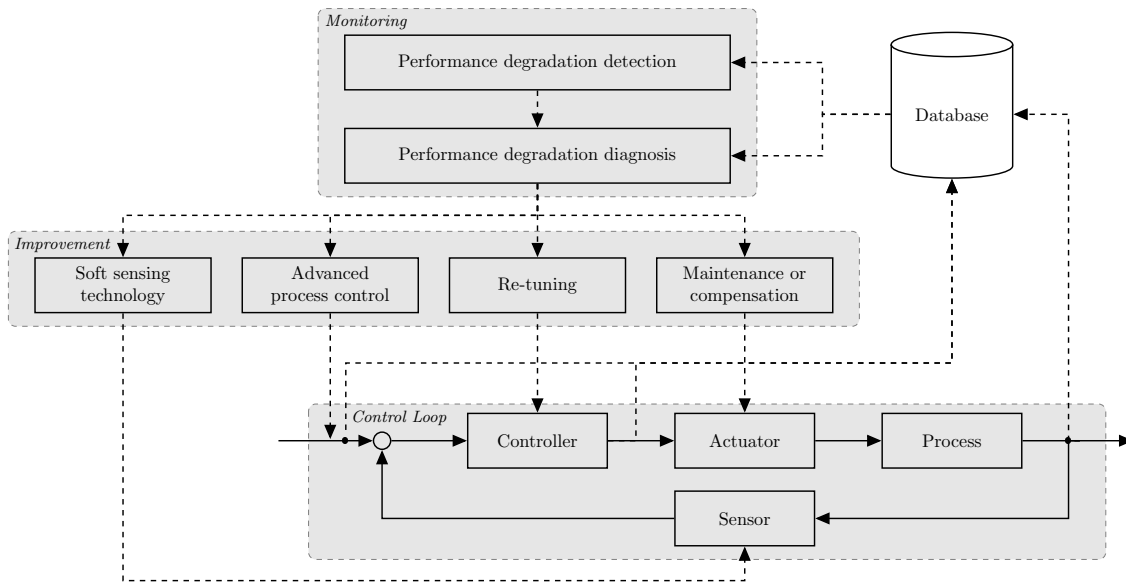


Figure 1.2: Process improvement based on control loop performance monitoring.

## 1.2 Goals

The overall goal of this thesis is to address a real need of the majority of process companies in the field of Process Supervision. This work covers areas such as real-time production monitoring, Proportional-Integral-Derivative type controller tuning, and soft sensor technology.

Based on the motivating considerations described in Section 1.1, the following goals were defined as targets to be accomplished in the present thesis:

- To review and develop system identification strategies fundamental in the development of methods in areas of monitoring and improvement of control loops performance.
- To review and systematize the current status of methods for the monitoring and improvement of control loops performance providing an insight into the assumptions and limitations of the main methods.
- To improve existent approaches for the detection of poor performance and the diagnosis of the underlying root causes;
- To improve existent PID controller tuning techniques to take into consideration aspects of poor performance such as weak controllers tuning and

problematic actuators;

- To contribute to broaden the use of soft sensors technology by identifying possible industrial application scenarios.

## 1.3 Overview

This thesis deals with the monitoring and improvement of the control loop performance in a context of large process plants. The present thesis structure is shown in Figure 1.3.

Chapter 2 concerns key system identification aspects that are used in the remaining chapters. Particularly, the closed-loop identification is addressed and its basic steps are outlined. In addition, techniques related to the model structure selection, parameter estimation, and the model validation are explored. Based on it, a comparison of commonly used model structures is performed and a hybrid modeling approach is presented as a means to obtain a compromise between model quality and computational burden.

Chapter 3 presents a review of the existing control loop performance monitoring methods, ranging from poor performance detection, disturbances type and specific malfunctions diagnosis to the performance improvements. A new method for the detection and characterization of multiple oscillations in industrial signals is proposed. Additionally, poor performance in control valves caused by stiction is addressed in two ways. First, a systematic taxonomy of the existing contributions, covering modeling, detection/quantification, and compensation of stiction is provided. Second, two new approaches for valve stiction detection are proposed.

In Chapter 4, the tuning of PID controllers is addressed as a strategy for performance optimization. Firstly, an introduction to PID control is given with a focus on the basic algorithm, the digital computer implementation, the PID controller parameters determination contemplating single and multiple loops, and the tuning methods automation. Then, a new optimization based method is proposed that considers the actuator problem explicitly in the tuning process.

In Chapter 5, the soft sensor technology is also addressed as a tool to improve the performance of an industrial control loop. Here, an overview of the tech-

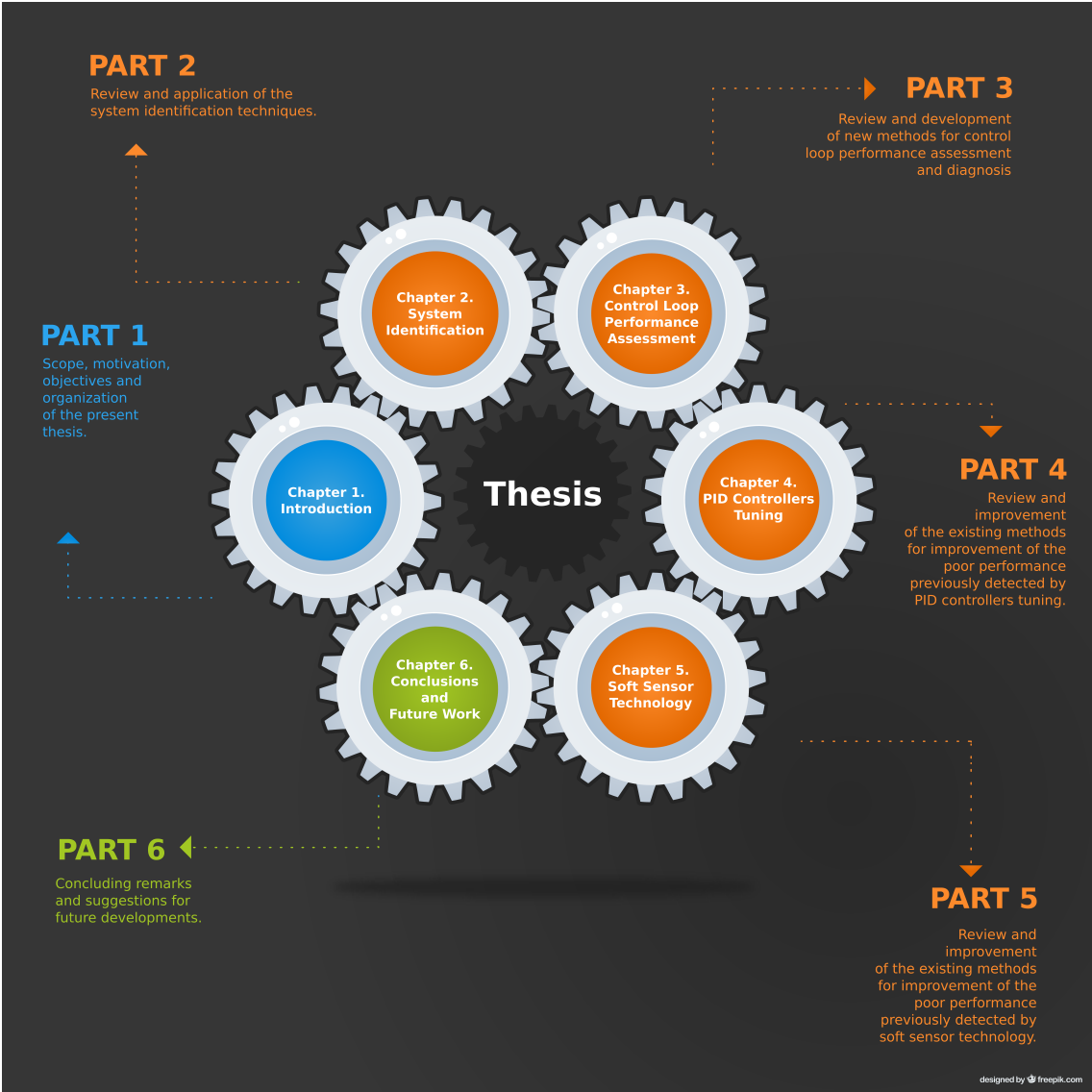


Figure 1.3: Thesis organization and overview.

niques used for soft sensor development and use is provided. In a case study, a soft sensor is developed for an industrial process in order to avoid measurement delays, to monitor the process in real-time, and to enable quick control actions.

Finally, Chapter 6 contains the main conclusions and contributions of the present thesis and outlines future work directions.



# Chapter 2

## System Identification

Various key factors associated with system identification are presented, particularly the closed-loop identification approaches as well as techniques related to the model structure selection, the parameter estimation, and the model validation. Two case studies cover applications to both a SISO and a MIMO systems and compare different model structures. Additionally, a hybrid strategy to cope with computationally heavy iterative calculations in the context of biodiesel industry is also proposed.

### 2.1 Definition

A mathematical model is a representation of a real system that allows to predict its behaviour in different scenarios. Model development should result in a compromise between realism and simplicity, i.e., the model should incorporate most of the real system significant features yet should not be so complex, difficult or even impossible to understand or experiment with (Maria, 1997). Besides, the number of model parameters should be carefully chosen so that it is possible to identify them with the existing measurements.

System identification is a term coined by Zadeh (1956) that deals with the construction of mathematical models based on the experimental observation of the system response to some stimuli. Its use is widespread across all engineering fields, from aerospace, to civil or health industries (Klein and Morelli, 2006; Pan, 2007; Eren-Oruklu et al., 2012). In the manufacturing and process industries, it is

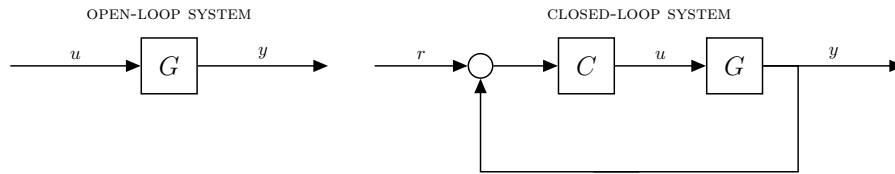


Figure 2.1: Block diagrams for open- and closed-loop systems, where  $u$  is the system input,  $y$  is the system output,  $r$  is the setpoint signal,  $C$  is the controller dynamics, and  $G$  is the system dynamics.

commonly used to obtain models for control purposes.

## 2.2 State-of-the-art

This section contains a general overview of the most important ideas and developments on system identification.

### 2.2.1 Closed-loop identification

The identification of systems may be performed using data from open- or closed-loop systems (Figure 2.1). Closed-loop identification is defined as the identification of the open-loop system dynamics while the controller tracks the reference signal (Söderström and Stoica, 1989; Ljung, 1999; Bakke, 2009). The advantages of closed-loop experiments are the fact that the control loop may remain in auto during the identification experiment, the presence of controllers may possibly linearise a non-linear plant behaviour around a relevant operating point helping to achieve accurate linear modelling, and the possibility of performing the identification while keeping the plant within safety and production limits. Some systems, such as biological and economic systems, require closed-loop identification because they contain feedback loops that may not be removed or opened. And, in some industrial systems, the open-loop dynamics may be unstable or so poorly damped that no identification experiment may be performed. However, there are some problems with the closed-loop identification that are not present in the open-loop case. For instance, the controller tuning may affect the information content of the data, even with the excitation persistence of the controller output. Also, the correlation between disturbances and input data as a consequence of



the feedback mechanism may compromise the estimation of the open-loop characteristics (Bakke, 2009).

When the control loop is closed, three common strategies are applied in order to estimate the open-loop characteristics: the direct, the indirect, and the joint input-output approaches (Forssell and Ljung, 1997, 1999; Forssell, 1999). Closed-loop identification using the direct approach consists of ignoring the feedback loop and performing the estimation using the inputs and output signals. The indirect approach assumes the controller model knowledge and uses the reference and the output signals to estimate the plant model. Finally, the joint input-output approach uses the three variables without other knowledge to perform the identification. It views the closed-loop system as a system with one input (the reference signal,  $r$ ) and two outputs (the input signal,  $u$ , and the output,  $y$ , signals) with the following transfer functions

$$G_{ry}(s) = \frac{GC}{1 + GC}, \quad (2.1)$$

$$G_{ru}(s) = \frac{C}{1 + GC}, \quad (2.2)$$

obtaining  $G$  from

$$G = \frac{G_{ry}}{G_{ru}}. \quad (2.3)$$

The direct approach is undoubtedly the simplest because it does not require the knowledge of the controller type and mode (open-loop, feedback, or feed-forward). The indirect approach requires a linear time-invariant controller and is strongly affected by non-linearities such as constraints and anti-reset windup. In addition, the estimates are often of higher order and some model reduction procedure may be needed. The joint input-output approach contains practical difficulty related to pole cancellation. In fact, although the denominators of  $G_{ry}$  and  $G_{ru}$  are theoretically equal and should cancel out when performing the  $G$  calculation, the presence of even small estimation errors of  $G_{ry}$  and  $G_{ru}$  may result in wrong system identification.

Several works assess the published system identification approaches using explicit criteria, such as bias and variance (Hof and Schrama, 1995; Van den Hof, 1998).

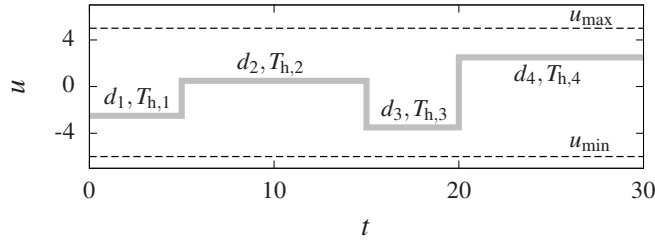


Figure 2.2: Amplitude modulated pseudo-random binary signal.

## 2.2.2 Fundamental steps

Identification is an iterative process that comprises the following four fundamental steps: data acquisition, selection of the model structure, parameter estimation, and model validation (Ikonen and Najim, 2001).

The acquisition of experimental data is a very important step that determines the information content in the data within the limits imposed by the process. The interval between consecutive samples (commonly referred as sampling period) must be small enough so that the significant process information is not lost. The collected signals must result from a persistently excited system to guarantee sufficient dynamical information (Ikonen and Najim, 2001). This is specially important in more complex high-order dynamic processes that usually are mildly perturbed around a nominal operation point resulting in insufficient information about the process non-linearity (Abonyi et al., 2000). The choice of the excitation signal is important for the achieved model quality. For instance, the amplitude modulated pseudo-random binary signal mentioned by (Deflorian and Zaglauer, 2011) is a periodic deterministic signal with amplitude values as free design parameters. It may be understood as a sequence of step functions defined by several design points  $d_i \in [u_{\min}, u_{\max}]$  and the dwell times  $T_{h,i}$  (Figure 2.2). In industrial practice, this signal may be not feasible because input variables jumps may lead to unsafe operating conditions (Ikonen and Najim, 2001). As an alternative, Deflorian and Zaglauer (2011) suggest to use ramps with a defined maximum allowable slope instead of jumps. In general, the acquired data needs some conditioning such as scaling and normalization to uniformize the magnitudes of the variables and filtering to remove noise. The data conditioning has very significant effects for multivariable systems (Ikonen and Najim, 2001).

The definition of the model structure starts with the selection of its inputs, outputs and internal components. This choice is a compromise between the accuracy of the predictions and the model complexity. The first- or second-order linear models are adequate in many cases and, consequently, are the first candidates. After the model parameters are determined in the parameter estimation phase, the goodness of the obtained model is assessed in the model validation phase. The validation methods verify model properties such as the fit accuracy, the generalization capability, and the computational efficiency. The selected model structure and their parameters may be readjusted in the model validation phase if its quality is not satisfactory (Ikonen and Najim, 2001).

### 2.2.3 Selection of the model structure

It should be noted that the model structure, complexity, and quality depend on its intended use. In most cases, it is desirable to use the simplest possible model form as long as it is capable of capturing the most important steady-state and dynamic characteristics of the process (Liu and Gao, 2011).

Process models may have various applications such as process design, control, optimization, or fault detection (Ikonen and Najim, 2001). For all these purposes, it formalizes the knowledge about the chemical and physical phenomena taking place in the process.

In process design, mathematical models may be a safe and inexpensive replacement for experiments on real processes. They may also help in the scale-up and process intensification. In process control, process model are used to predict the output variable to determine the optimal control moves. Besides, PID controller tuning requires a mathematical model of the process (Åström and Hägglund, 2006). In process optimization, the model may be used for simulating the process behaviour in different operating conditions. It may also be used within a numerical optimizer in order to meet specific plant objectives. In this context, the model may integrate the operator decision support system or be used to the personnel training. In fault detection, anomalies of the plant may be continuously monitored by comparing the model prediction with the measured variables. Other application is the monitoring of variables that are not directly available through existing measurements or are subject to long measurement delays.

These models are also called soft sensors.

Several approaches and techniques are available for deriving mathematical models. Standard approaches include first-principle modeling and black-box modeling (Ikonen and Najim, 2001). The first-principle (or mechanistic) modeling involves the use of physical laws and relationships that determine the system behaviour. The model structure incorporates all physical insight about the process and its variables and parameters have physical interpretation. This direct modelling may often be impossible because the knowledge of the system is incomplete, or because the system properties may change in an unpredictable way. Besides, this type of modelling may be very time consuming and may lead to unnecessarily complex models (Ikonen and Najim, 2001). For these reasons, the most common approach used in identification is the black-box (or experimental) in which the models are obtained with no a priori information available. Instead, the functional form of relations between the inputs and outputs and the values of parameters are determined from the experimental data (Ikonen and Najim, 2001). Finally, the grey-box approach utilizes physical insight into the observed system but with lesser complexity than that of the first-principle approach (Ikonen and Najim, 2001).

A typical modelling approach is the use of linear relationships among the model variables. Such models are simple, flexible, robust, and efficient variables. In what concerns linear systems, if they present an explicit dependence on time, they are called linear time variant. A model that is linear and does not depend explicitly on time is said a linear time invariant model, usually abbreviated as LTI model (Ljung, 1999).

Although linear models are the most common way of describing a dynamical system, it is often needed to employ more complex descriptions because most industrial processes are non-linear (Ikonen and Najim, 2001).

Some of the modelling methodologies that fall into the aforementioned categories are depicted in Figure 2.3 and described in the following subsections.

### ■ Finite impulse response models

In linear time-invariant stable processes, the dynamics is uniquely characterized by the impulse response that will tend to zero along time. This is the base of finite

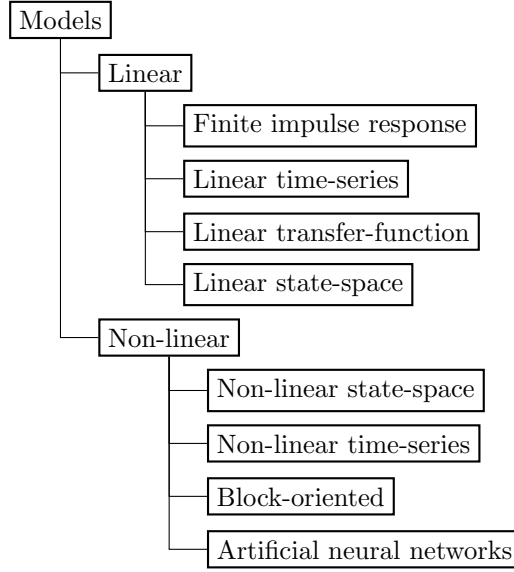


Figure 2.3: Classification of models by their linearity.

impulse response (FIR) models (also designated by Markov parameter models). The FIR model of a single input single-output process is given by

$$y(t/\Delta t) = \sum_{k=1}^{n_\theta} \theta(k) u(t/\Delta t - k) + \epsilon(t/\Delta t) + d(t/\Delta t), \quad (2.4)$$

where  $\theta(k)$  are the model parameters,  $u(t/\Delta t - k)$  the input variable at time  $t/\Delta t - k$ ,  $\epsilon(t/\Delta t)$  the model residual,  $d(t/\Delta t)$  the process disturbance, and  $n_\theta$  the truncation number. In the matrix form for all the horizon, it is possible write that

$$\mathbf{y} = \mathbf{\Phi} \boldsymbol{\theta} + \boldsymbol{\epsilon} + \mathbf{d}, \quad (2.5)$$

where  $\mathbf{y} \in \mathbb{R}^{N \times 1}$  is the output variable prediction,  $\mathbf{\Phi} \in \mathbb{R}^{N \times n_\theta}$  is a Hankel matrix containing the input variable,  $\boldsymbol{\theta} \in \mathbb{R}^{n_\theta \times 1}$  is the parameter vector,  $\boldsymbol{\epsilon} \in \mathbb{R}^{N \times 1}$  is the model residual,  $\mathbf{d} \in \mathbb{R}^{N \times 1}$  is the process disturbance (white or coloured noise), and  $N$  is the number of points to be predicted (Zhu, 2001; Ikonen and Najim, 2001; Dayal and MacGregor, 1996).

In cases where the noise is coloured, the process disturbance model may be simultaneously identified with the FIR model. The choice of the disturbance model structure may be critical. However, Ljung (1999) proved that FIR models will

converge to the correct solution even in the case of coloured open-loop data and stationary disturbances (Dayal and MacGregor, 1996).

FIR models are advantageous because they require little prior knowledge of the process (such as the model order and structure) and its estimates are statistically unbiased (the estimation expectation equals the true value) and consistent (the estimate tends to the true value when the number of samples tends to infinity). However, they may not model unstable processes and they often require a very high truncation number which increases the number of parameters to be estimated (Zhu, 2001; Ikonen and Najim, 2001).

### ■ Time series models

Although there are numerous time series model structures, the most commonly used in practice are the linear black-box structures that are variants of a generalized model proposed by (Ljung, 1999, Equation 4.34) and given by

$$A(q) y(t) = q^{-\tau} \frac{B(q)}{F(q)} u(t) + \frac{C(q)}{D(q)} \xi(t), \quad (2.6)$$

where  $y(t)$  and  $u(t)$  are the process output and input, respectively,  $\xi(t)$  is the system disturbance (usually assumed to be noise identically distributed with zero mean and finite variance), and  $\tau \geq 1$  is the time delay of the discretized process expressed as an integer multiple of the sampling time  $\Delta t$  added with the unity, that is,

$$\tau = \frac{\theta}{\Delta t} + 1. \quad (2.7)$$

The polynomial functions  $A(q)$ ,  $B(q)$ ,  $C(q)$ ,  $D(q)$ , and  $F(q)$  are polynomials<sup>1</sup> defining deterministic and stochastic parts of the system and are given by

$$A(q) = \sum_{i=1}^{\infty} a_i q^{-i} \quad (2.8)$$

$$B(q) = b_0 + \sum_{i=1}^{\infty} b_i q^{-i}, \quad (2.9)$$

---

<sup>1</sup>The backwards-shift (or unit delay) operator  $q$  is defined as  $q^{-n} f(k) = f(k - n)$ .

$$C(q) = 1 + \sum_{i=1}^{\infty} c_i q^{-i} \quad (2.10)$$

$$D(q) = 1 + \sum_{i=1}^{\infty} d_i q^{-i} \quad (2.11)$$

$$F(q) = 1 + \sum_{i=1}^{\infty} f_i q^{-i}, \quad (2.12)$$

usually approximated by

$$A(q) = \sum_{i=1}^{N_A} a_i q^{-i} \quad (2.13)$$

$$B(q) = b_0 + \sum_{i=1}^{N_B} b_i q^{-i} \quad (2.14)$$

$$C(q) = 1 + \sum_{i=1}^{N_C} c_i q^{-i} \quad (2.15)$$

$$D(q) = 1 + \sum_{i=1}^{N_D} d_i q^{-i} \quad (2.16)$$

$$F(q) = 1 + \sum_{i=1}^{N_F} f_i q^{-i}, \quad (2.17)$$

where  $N_A, N_B, N_C, N_D$  and  $N_F$  are polynomial orders.

Different models may be obtained from this structure. The auto-regressive (AR) and the moving-average (MA) models are the simplest forms of time series. The AR model form is defined by

$$A(q) y(t) = \xi(t). \quad (2.18)$$

Figure 2.4 depicts the signal flow of the model.

In contrast, the moving-average (MA) model has the form

$$y(t) = C(q) \xi(t). \quad (2.19)$$

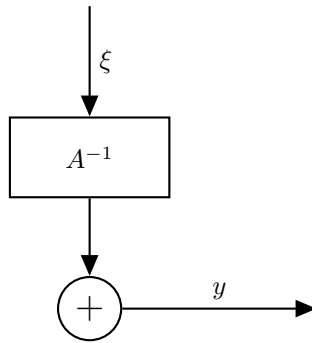


Figure 2.4: AR model structure.

When AR and MA representations are combined, it results the auto-regressive moving-average (ARMA) model that may be represented using the general form

$$A(q) y(t) = C(q) \xi(t).$$

In a control context, the input variable influences the process behaviour. To include these exogenous inputs to the system in a time series model the later model may be extended to

$$A(q) y(t) = q^{-\tau} B(q) u(t) + C \xi(t). \quad (2.20)$$

This form is called as auto-regressive moving-average with exogenous input (ARMAX) model.

Sung et al. (2009) refer that the most used forms in process systems engineering are the auto-regressive exogenous (ARX) model and the output error (OE) model. Derived from the generalized model (2.6), the ARX model is defined as

$$A(q) y(t) = q^{-\tau} B(q) u(t) + \xi(t). \quad (2.21)$$

Using the ARX model, the output variable  $y(t)$  may be predicted only one step ahead because its estimation depends on the past process outputs.

The OE model has the following structure:

$$y(t) = q^{-\tau} \frac{B(q)}{F(q)} u(t) + \xi(t). \quad (2.22)$$



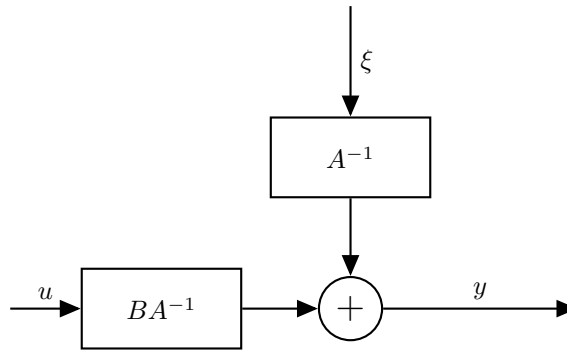


Figure 2.5: ARX model structure.

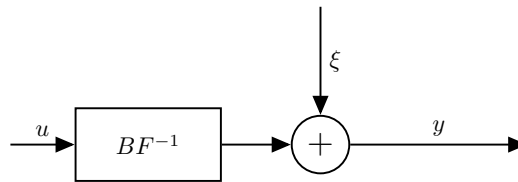


Figure 2.6: OE model structure.

Figure 2.6 shows the signal flow of this model. Because the determination of  $y(t)$  at the current time depends exclusively on the past model output, it is possible to estimate the model output in the future if the process inputs are known.

Another form frequently mentioned in literature is the Box-Jenkins model that is structured as

$$y(t) = q^{-\tau} \frac{B(q)}{F(q)} u(t) + \frac{C(q)}{D(q)} \xi(t). \quad (2.23)$$

Its signal flow is depicted in Figure 2.7.

### ■ Transfer-function models

Dynamic models derived from physical principles typically consist of one or more ordinary differential equations (ODE). Therefore, this kind of equations are also good candidate models for system identification purposes. The first-order (FO), first-order with time delay, and second-order (SO) models are very useful to design and implement process controllers.

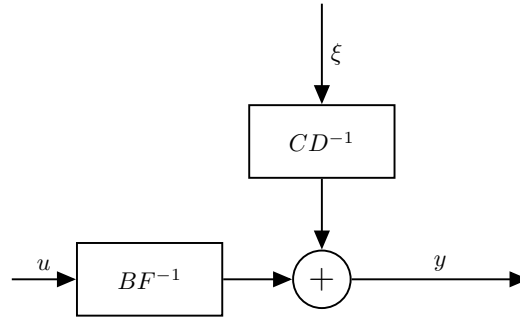


Figure 2.7: Box-Jenkins model structure.

The first-order (FO) model is defined as

$$\tau \dot{y}^*(t) + y^*(t) = K_p u^*(t), \quad (2.24)$$

where  $y^*$  and  $u^*$  are the output observed variable and the input variable, respectively, both expressed via deviation variables,  $K_p$  is the static gain and  $\tau$  is the time constant. The deviation variables  $y^*$  and  $u^*$  are related to the original variables  $y$  and  $u$  through a simple translation of the initial steady-state  $\bar{y}$  and  $\bar{u}$ , respectively, that is,

$$y^* = y - \bar{y} \quad (2.25)$$

and

$$u^* = u - \bar{u}. \quad (2.26)$$

The static gain,  $K_p$ , represents how much the process output changes, from one steady-state to another for a unitary variation of the process input while the time constant,  $\tau$ , represents how fast the process responds to a change in the process input.

The second-order (SO) model is mathematically described by

$$\ddot{y}^*(t) + 2\xi\omega \dot{y}^*(t) + \omega^2 y^* = K_p \omega^2 u^*(t), \quad (2.27)$$

where  $\xi$  is the damping factor that determines the oscillatory behaviour of the system,  $\omega$  is the undamped natural frequency, and  $K_p$  is the gain of the system.

In simple processes, each output variable depends essentially on a single input variable. They can be seen as *single-input single-output* (SISO) systems. How-

ever, a large class of processes exhibits interaction among variables, i.e., each output variable is dependent on a subset of the input variables. These latter processes are regarded as *multiple-input multiple-output* (MIMO) systems. The common industrial practice is to assume that there is no interaction between different control loops or to design controllers in a way that weakens the interaction. However, such approaches may result in suboptimal plant performance. Therefore, multivariable controller tuning and, thus, the multivariable system identification which is the subject of this work, have a big practical importance.

### ■ State-space models

State-space models provide a compact and useful representation of a set of linear ODEs and can be generally written as

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{A} \mathbf{x}(t) + \mathbf{B} \mathbf{u}(t) \\ \mathbf{y}(t) &= \mathbf{C} \mathbf{x}(t) + \mathbf{E} \mathbf{u}(t)\end{aligned}\quad (2.28)$$

where  $\mathbf{x}(t)$  is the state vector,  $\mathbf{u}(t)$  is the input variables vector,  $\mathbf{y}(t)$  is the output vector of observed variables, and parameters  $\mathbf{A}$ ,  $\mathbf{B}$ ,  $\mathbf{C}$ , and  $\mathbf{E}$  are constant matrices of sizes  $n_x \times n_x$ ,  $n_x \times n_u$ ,  $n_y \times n_x$  and  $n_y \times n_u$ , respectively. Typically, the observed variables are a subset of the state variables or a linear combination of them (Seborg et al., 2010) and thus  $\mathbf{E}$  is the null matrix in such a case.

System (2.28) may accommodate linear first-order ODEs directly and higher order equations after a pre-treatment step in which higher order dynamics is represented by a set of first-order equations (Guillet et al., 2011; Salimbahrami and Lohmann, 2006). In the particular case of a second-order MIMO system, given in the form

$$\begin{aligned}\mathbf{M} \ddot{\mathbf{z}}(t) + \mathbf{D} \dot{\mathbf{z}}(t) + \mathbf{K} \mathbf{z}(t) &= \mathbf{F} \mathbf{u}(t) \\ \mathbf{y}(t) &= \mathbf{H} \mathbf{z}(t)\end{aligned}\quad (2.29)$$

where  $\mathbf{z} \in \mathbb{R}^{n_z}$  and  $\mathbf{M}$ ,  $\mathbf{D}$ ,  $\mathbf{K}$  ( $\in \mathbb{R}^{n_z \times n_z}$ ),  $\mathbf{F}$  ( $\in \mathbb{R}^{n_z \times n_u}$ ),  $\mathbf{H}$  ( $\in \mathbb{R}^{n_y \times n_z}$ ) are constant

matrices, it is possible to write equivalently that

$$\begin{aligned} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{M} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{z}}(t) \\ \ddot{\mathbf{z}}(t) \end{bmatrix} &= \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{K} & -\mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{z}(t) \\ \dot{\mathbf{z}}(t) \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{F} \end{bmatrix} \mathbf{u}(t), \\ \mathbf{y}(t) &= \begin{bmatrix} \mathbf{H} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{z}(t) \\ \dot{\mathbf{z}}(t) \end{bmatrix} \end{aligned}, \quad (2.30)$$

where  $\mathbf{I}$  and  $\mathbf{0}$  represent the identity matrix and the zero matrix of appropriate sizes, respectively. Therefore,

$$\begin{aligned} \underbrace{\begin{bmatrix} \dot{\mathbf{z}}(t) \\ \ddot{\mathbf{z}}(t) \end{bmatrix}}_{\mathbf{x}(t)} &= \underbrace{\begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{M}^{-1}\mathbf{K} & -\mathbf{M}^{-1}\mathbf{D} \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} \mathbf{z}(t) \\ \dot{\mathbf{z}}(t) \end{bmatrix}}_{\mathbf{x}(t)} + \underbrace{\begin{bmatrix} \mathbf{0} \\ \mathbf{M}^{-1}\mathbf{F} \end{bmatrix}}_{\mathbf{B}} \mathbf{u}(t), \\ \mathbf{y}(t) &= \underbrace{\begin{bmatrix} \mathbf{H} & \mathbf{0} \end{bmatrix}}_{\mathbf{C}} \underbrace{\begin{bmatrix} \mathbf{z}(t) \\ \dot{\mathbf{z}}(t) \end{bmatrix}}_{\mathbf{x}(t)} \end{aligned}, \quad (2.31)$$

as explained by Brásio et al. (2012).

### ■ Non-linear state-space models

Non-linear state-space models are represented in the form

$$\dot{x}(t) = f(x(t), u(t), w(t), \theta) \quad (2.32)$$

$$y(t) = h(x(t), u(t), v(t), \theta), \quad (2.33)$$

where  $w(t)$  and  $v(t)$  are disturbances assumed to be of known form (e.g., Gaussian),  $\theta$  is the vector of unknown parameters, and  $f(\cdot)$  and  $h(\cdot)$  are non-linear functions (Ljung, 1999).

### ■ Non-linear time series models

Non-linear time series models extend the linear time series models to the non-linear case. Considering an ARX model, the respective non-linear ARX model has the structure

$$y(t) = f(\varphi(t), \theta) + \xi(t), \quad (2.34)$$

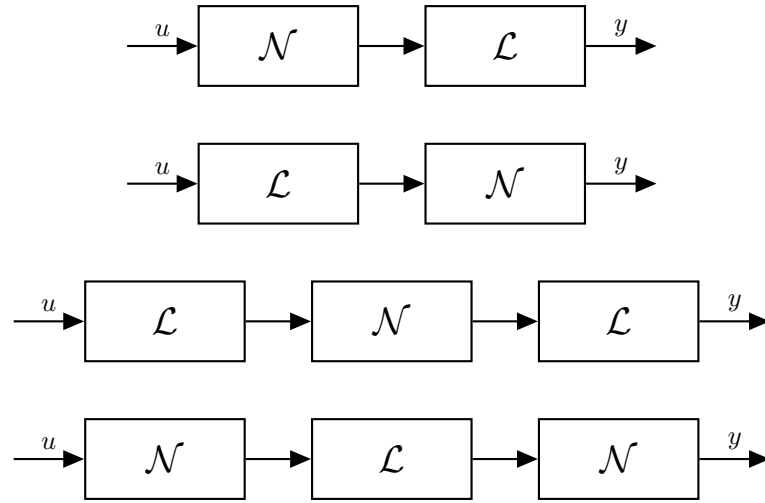


Figure 2.8: Block-oriented non-linear models. First line: Hammerstein model, second line: Wiener model, third line: Wiener-Hammerstein model, fourth line: Hammerstein-Wiener model.

where the function  $f(\cdot)$  is a flexible non-linearity estimator with parameters that do not need to have physical significance (Lyzell, 2009, Section 3.2.3).

### ■ Block-oriented models

Simple structures may be constructed through the block-oriented models (Ljung, 1999), where each of them represent an individualized part of the whole system. The block-oriented models are connections of static (memoryless) non-linear function blocks to dynamical linear blocks. The simplest model consists in putting a non-linear gain before a linear block. This scheme is commonly called as Hammerstein model or  $\mathcal{N} - \mathcal{L}$  model, where  $\mathcal{N}$  stands for non-linear operator and  $\mathcal{L}$  linear operator. When reversing the order of the blocks, it results a structure called Wiener model or  $\mathcal{L} - \mathcal{N}$  model. More complex block-oriented configurations, as shown in Figure 2.8, may provided useful structures for modelling the non-linearities present in a system.

### ■ Artificial neural networks

Artificial neural networks are used in many engineering applications for predicting variables of complex systems (Haykin, 1998; Chaturvedi, 2008; Du et al.,

2011). Their use allows the simulation of physical phenomena without explicit mechanistic formulation to describe the relationships between the variables (Du et al., 2011).

Feedforward back-propagation are the simplest and the most used type of neural networks that are typically composed by three layer types of neurons or nodes (one input layer, one or more intermediate layers and one output layer). They are considered static because their outputs depend only on the current input variables and constants. The absence of further information (feedback) ensures the stability of the model (Chaturvedi, 2008).

The number of intermediate layers can vary increasing the prediction capacity of the network, which proves particularly useful in problems with a large number of input variables. However, an increase in the number of these layers also contributes to the over-training of the network due to the large number of parameters to determine, that is, it can lead to the over-fitting of the network (Bishop, 1995; Chaturvedi, 2008), apart from increasing exponentially its learning time.

Intermediate and output neurons are structured by an aggregate function and an activation function. Commonly, the sum function is used as an aggregation function in the neural networks structure. Regarding the activation function, the most common are the linear, the sigmoid and the hyperbolic tangent functions (Chaturvedi, 2008). Usually, the hyperbolic tangent and the linear function are chosen for intermediate and output layers.

Each neuron is directly connected to the neurons of the adjacent layers. Each link is assigned a weight that represents the degree of relationship between the two neurons involved. Mathematically, one can write (Bishop, 1995; Chaturvedi, 2008)

$$I_j = f_I \left( \sum_{i=1}^{n_X} w_{ij} \cdot X_i + \theta_j \right), \quad j = 1, \dots, n_I \quad (2.35)$$

and

$$Y_k = f_Y \left( \sum_{j=1}^{n_I} W_{jk} \cdot I_j + \Gamma_k \right), \quad k = 1, \dots, n_Y, \quad (2.36)$$

where  $n_X$  is the number of input neurons,  $n_I$  is the number of intermediate neurons,  $n_Y$  is the number of output neurons,  $X_i$  is the input neuron  $i$ ,  $I_j$  is the intermediate neuron  $j$ ,  $Y_k$  is the output neuron  $k$ ,  $w_{ij}$  is the weight of the input

neuron  $i$  relatively to the intermediate neuron  $j$ ,  $W_{jk}$  is the weight of the intermediate neuron  $j$  relatively to the output neuron  $k$ ,  $\theta_j$  is the bias of the intermediate neuron  $j$ ,  $\Gamma_k$  is the bias of the output neuron  $k$ ,  $f_I(\cdot)$  is the activation function of the intermediate neurons and  $f_Y(\cdot)$  is the activation function of the output neurons.

The activation function  $f_I(\cdot)$  is usually defined through the hyperbolic tangent since it allows a faster convergence of the training algorithm (Bishop, 1995). As for the activation function to the output layer,  $f_Y(\cdot)$ , it is generally a linear function (Chaturvedi, 2008). Mathematically,

$$\text{Hyperbolic tangent : } f_I(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}, \quad -1 < f_I < 1 \quad (2.37)$$

$$\text{Linear : } f_Y(z) = z, \quad -\infty < f_Y < \infty \quad (2.38)$$

where  $z$  is a generic variable.

In a typical configuration, the continuous and differentiable function with predictive objectives which is generated by the neural network is defined in the vectorial form by

$$\mathbf{Y}(\mathbf{P}, \mathbf{X}) = \mathbf{W} \cdot \tanh(\mathbf{w}^\top \cdot \mathbf{X} + \boldsymbol{\theta}) + \boldsymbol{\Gamma}, \quad (2.39)$$

where  $\mathbf{P}$  represents the set of matrix parameters  $\mathbf{w} \in \mathbb{R}^{n_x \times n_I}$ ,  $\mathbf{W} \in \mathbb{R}^{n_Y \times n_I}$ ,  $\boldsymbol{\theta} \in \mathbb{R}^{n_I \times 1}$  and  $\boldsymbol{\Gamma} \in \mathbb{R}^{n_Y \times 1}$ ;  $\mathbf{X} \in \mathbb{R}^{n_x \times 1}$  represents the vector of input neurons and  $\mathbf{Y} \in \mathbb{R}^{n_Y \times 1}$  the vector of output neurons.

Considering a dataset with  $m$  points  $\{(\mathbf{X}_1, \mathbf{Y}_1), \dots, (\mathbf{X}_i, \mathbf{Y}_i), \dots, (\mathbf{X}_m, \mathbf{Y}_m)\}$ , the neural network training is the process of determination of the parameters so that, for the input  $\mathbf{X}_i$ , the estimate of the output variables  $\hat{\mathbf{Y}}_i$  should match as close as possible the values  $\mathbf{Y}_i$  (Yegnanarayana, 2004). Such optimization problem corresponds to the minimization of the average square error (MSE), that is,

$$\min_{\mathbf{P}} F(\mathbf{P}) = \frac{1}{m} \sum_{i=1}^m \mathbf{e}_i^\top \mathbf{e}_i,$$

where  $\mathbf{e}_i = \mathbf{Y}_i - \hat{\mathbf{Y}}_i(\mathbf{P}, \mathbf{X})$ . The Levenberg-Marquardt algorithm is commonly used to solve this problem because of its high performance and robustness even in cases of strong ill-conditioned problems (Sjöberg, 2005).

### 2.2.4 Parameter estimation

Parameter estimation is the process of computing efficiently numerical values for the parameters of a known mathematical model of observations with the appropriate tools (Beck and Arnold, 1977; Zhang, 1997). Usually, it is seen as an optimization problem which minimizes a cost function consisting of a sum of squared prediction errors (Ikonen and Najim, 2001).

Parameter estimation has a long history in deterministic methods in the sense that no statistical assumptions are made with respect to the observations errors in the measured parameters. For instance, the gradient-based methods are dominant because they have shown to be efficient in practice. Their main disadvantage is that they may get stuck in local minima of the cost function. These methods consider that errors in the observations are absent what has motivated the development of statistical parameter estimation methods (Beck and Arnold, 1977), although the practical implementations are often inefficient (Ikonen and Najim, 2001) and complex.

#### ■ Least squares method

Developed by Karl Gauss when he was trying to characterize the motions of planets and comets using telescopic measurements, the least squares method is essential in systems and control engineering (Ikonen and Najim, 2001). Considering a linear regression model with the form

$$\mathbf{y} = \Phi\boldsymbol{\theta} + \boldsymbol{\xi}, \quad (2.40)$$

where  $\boldsymbol{\theta}$  is the column vector of parameters to be estimated,  $\mathbf{y}$  is the column vector of observations,  $\Phi$  is the matrix of plant measurements, and  $\boldsymbol{\xi}$  is the column vector of system noise. The method obtains the estimate  $\hat{\boldsymbol{\theta}}$  of the model parameters  $\boldsymbol{\theta}$  that minimizes the sum of the squared residuals defined as

$$J(\boldsymbol{\theta}) = \frac{1}{N}(\mathbf{y} - \Phi\boldsymbol{\theta})^\top \boldsymbol{\alpha}(\mathbf{y} - \Phi\boldsymbol{\theta}), \quad (2.41)$$

where  $N$  is the number of observations and  $\boldsymbol{\alpha}$  is the weighting factors matrix. For the ordinary least-squared method, the weighting factors matrix is the identity



matrix, while for weighted least squares method the elements of the diagonal assume values different of ones. In a more compact form, the least squares method is represented by

$$\hat{\boldsymbol{\theta}} = [\boldsymbol{\Phi}^\top \boldsymbol{\Phi}]^{-1} \boldsymbol{\Phi}^\top \mathbf{y}, \quad (2.42)$$

where the Hessian matrix,  $\mathbf{H} = \boldsymbol{\Phi}^\top \boldsymbol{\Phi}$ , must be positive definite.

Hence, a linear regression model

$$\hat{\mathbf{y}} = \boldsymbol{\Phi} \hat{\boldsymbol{\theta}}, \quad (2.43)$$

is identified.

### ■ Numerical optimization

Numerical optimization is a very used tool to find the parameters of a given model. Mathematically, optimization is the minimization or maximization of a function subject to constraints on its variables. In the context of parameter estimation, the optimization problem may be written as

$$\underset{\boldsymbol{\theta}}{\text{minimize}} \quad J(\boldsymbol{\theta}) \quad (2.44a)$$

subject to

$$c_i(\boldsymbol{\theta}) = 0, \quad i \in \mathcal{E} \quad (2.44b)$$

$$c_i(\boldsymbol{\theta}) \geq 0, \quad i \in \mathcal{I}, \quad (2.44c)$$

where  $J$  denotes the objective function,  $\boldsymbol{\theta}$  is the model parameter vector to be estimated, and  $c_i$  are the vector of constraints that the parameters must satisfy.

A good numerical optimization algorithm should be robust, efficient, and accurate. The robustness of an algorithm is defined by its capacity to perform well on a wide variety of problems for different and reasonable choices of the initial values of the variables. An algorithm must also be efficient not requiring too much computer time or storage. And an accurate algorithm identifies with precision the solution without being overly sensitive to errors in the data. These goals may be very conflictive because, for instance, a rapidly convergent method may require too much storage on large problems or a robust algorithm may be too slow.

Optimization algorithms begin with an initial guess of the optimal values of the parameters  $\theta_n$  and generate improved estimates  $\theta_{n+1}$  until a set of parameters that allows a satisfactory representation of the system behaviour is found. The strategy used to move from one iteration to the next distinguishes the different algorithms. The two fundamental iterative strategies for moving the iterations are the line search and the trust region. In both iterative strategies, the next iteration is given by

$$\theta_{n+1} = \theta_n + \alpha_n d_n \quad (2.45)$$

where  $\alpha_n$  is the step length and  $d_n$  is the search direction. In the line search strategy, the algorithm fixes a direction  $d_n$  and then searches the appropriate distance  $\alpha_n$  to move along solving the minimization problem

$$\underset{\alpha_n > 0}{\text{minimize}} \quad J(\theta_n + \alpha_n d_n). \quad (2.46a)$$

Because the exact minimization is very expensive, the line search strategy generates a limited number of trial step lengths and finds one with the lower objective function value. In opposition, in the trust region strategy, a region with radius  $\alpha_n$  is firstly defined and a candidate for the direction  $d_n$  is obtained solving the problem

$$\underset{d_n}{\text{minimize}} \quad \tilde{J}(\theta_n + d_n), \quad (2.47a)$$

where  $\tilde{J}$  is a linear approximation of the actual objective function  $J$  with similar behaviour near the current point  $\theta_n$ . The trust region is usually a ball defined by the condition  $\|d_n\|_2 \leq \Delta_n$ , with  $\Delta_n > 0$ . The new  $\theta_{n+1} = \theta_n + d_n$  must lie inside this trust region. When the candidate solution does not produce a sufficient decrease in the objective function, it means that the trust region is too large and must be shrunk reducing  $\Delta_n$  and the sub-problem is then re-solved.

### Newton method

The Newton method, also called Newton-Raphson method, is a very known approach used to find the roots of a function. As reviewed by Nocedal and Wright (1999) and Stewart (2010), consider the tangent line to a given point  $(\theta_n, J(p_n))$ . The idea behind the Newton method is that the  $x$ -axis intercept

of the tangent  $\theta_{n+1}$  is close to the root and may be calculated through

$$\theta_{n+1} = \theta_n - \frac{J(\theta_n)}{\nabla J(\theta_n)}, \quad (2.48)$$

where  $\nabla J(\theta_n)$  is the gradient of the objective function. This equation may be reformulated for the line search iterative strategy as

$$\theta_{n+1} = \theta_n + \alpha_n d_n \quad (2.49)$$

$$d_n = -B_n^{-1} \cdot \nabla J(\theta_n), \quad (2.50)$$

where  $B_n = \nabla^2 J(\theta_n)$  is the exact Hessian. The Newton method is very simple to apply and has a good local quadratic convergence. However, it is not guaranteed that it will converge when the initial estimate of the parameters is too far from the exact root. Also, convergence problems may occur when the tangent line becomes parallel to the  $x$ -axis.

### Gauss-Newton algorithm

The Gauss-Newton method is an algorithm to minimize non-linear objective functions exploiting the structure of the Jacobian. Reviewed by Dennis and Schnabel (1983); Nocedal and Wright (1999), the method consists in a modification of the Newton method with line search that approximates the exact Hessian usually calculated through

$$\nabla^2 J(\theta_n) = \mathbf{Jac}(\theta_n)^\top \mathbf{Jac}(\theta_n) + \sum_{j=1}^m r_j(p_n) \nabla^2 r_j(p_n), \quad (2.51)$$

by

$$\nabla^2 y(p_n) \approx \mathbf{Jac}(\theta_n)^\top \mathbf{Jac}(\theta_n), \quad (2.52)$$

where  $\mathbf{Jac}(\theta_n)$  is the Jacobian and  $r_j(p_n)$  are the residuals.

This approximation is advantageous because the exact second derivatives may sometimes be challenging to compute. Besides, this term is, in some situations, much more significant. For zero-residual problems, the method is local and quadratically convergent. And for problems that have reasonably small residuals, it is quick and locally convergent. On the contrary,

the Gauss-Newton is slow, local and linearly convergent on problems that are very non-linear or have reasonably large residuals. It also suffers from occasional non-convergence if the Jacobian does not have full column rank.

### Levenberg-Marquardt method

The Levenberg-Marquardt method, also known as the damped least squares method, locates the minimum of a function expressed as the sum of squares of non-linear real-valued functions as reviewed by Levenberg (1944); Nocedal and Wright (1999). Consisting in a linear combination of the steepest descent and the Gauss-Newton method, it has become a standard method for non-linear least squares problems widely adopted in several subjects. While the Gauss-Newton method is a modified Newton method using line search, the Levenberg-Marquardt method is derived by replacing the line search with the trust region iterative strategy. The usage of this strategy avoids the disadvantage of the Gauss-Newton method related to the cases which are rank-deficients or nearly so.

For a spherical trust region of radius  $\Delta_n$  and considering the linear function  $\tilde{J}_n$  that approximates the actual objective function  $J$  with similar behaviour near the current point  $\theta_n$ , the Levenberg-Marquardt method solves in each iteration the following problem

$$\underset{d_n}{\text{minimize}} \quad \frac{1}{2} \|\text{Jac}(\theta_n) d_n + r_n\|^2 \quad (2.53a)$$

subject to

$$\|d_n\| \leq \Delta_n, \quad (2.53b)$$

where  $\|\cdot\|$  is the Euclidean norm.

The algorithm combines the advantage of the steepest descent method related to the operating stability with the accelerated convergence in the minimum vicinity of the Newton method. However, there are also two important disadvantages. Firstly, the initial estimates of the model parameters must be close (within one order of magnitude) to the true values in unfavourable cases. Secondly, the method does not deal very well with physical bounds imposed on model parameters resulting in infeasibility.

### ■ Bayesian method

The classical approach to estimate parameters assumes an unknown but objectively fixed parameter  $\theta$ . Instead, the Bayesian approach (Box and Tiao, 1973; Bolstad, 2004; Nielsen, 2009) to parameter estimation uses probabilities to represent the uncertainty fixing the data and assuming several values for  $\theta$ . The Bayesian method uses distribution models to estimate parameters.

Consider the dataset  $d$  from which the parameters  $\theta$  will be estimated. The Bayes rule used in the Bayesian method is fundamentally composed by four distributions:

- the posterior probability distribution: the probability distribution of a particular set of parameters is given by the observed data,  $P(\theta|d)$ ;
- the likelihood function or sampling distribution: the probability distribution that a given set of parameters would have generated the observed data,  $P(d|\theta)$ ;
- the prior probability distribution over  $\theta$ : the probability distribution that contains the knowledge of the unknown parameters before any data have been observed,  $P(\theta)$ ; and
- the evidence distribution: being independent of parameters, it measures the probability that a particular realization is observed,  $P(d)$ .

In the Bayesian method, parameter estimation means seeking the posterior distribution through the usage of the Bayes theorem

$$P(\theta|d) = \frac{P(d|\theta) P(\theta)}{P(d)}, \quad (2.54)$$

incorporating considerable statistical information in form of the likelihood, the prior and the evidence distributions. Since the evidence is independent of the parameters, the Bayes theorem is often written as

$$P(\theta|d) \propto P(d|\theta) P(\theta). \quad (2.55)$$

Then the posterior distribution may be employed in the context of various estimation criteria, such as the minimum mean square error (MMSE) estimation or

the maximum a posteriori (MAP) estimation. The Bayesian estimator under the MMSE criterion is given by

$$\hat{\theta}_{\text{MMSE}} = \text{E}(\theta|d) \quad (2.56)$$

$$= \int_{\theta} \theta \cdot P(\theta|d) \, d\theta, \quad (2.57)$$

while under the MAP criterion is given by

$$\hat{\theta}_{\text{MAP}} = \arg \max_{\theta} P(\theta|d). \quad (2.58)$$

For time-varying systems, the first-order recursive Markov chain

$$\theta(k+1) = a \cdot \theta(k) + \Delta\theta(k), \quad (2.59)$$

is a particular and convenient stochastic model to update the model parameters values (Enzner, 2010). In (2.59),  $0 < a < 1$  represents the transition coefficient and  $\Delta\theta(k)$  the independent process noise with zero mean and covariance  $\sigma_{\Delta\theta}^2 = \text{E}(\Delta\theta(k) \cdot \Delta\theta^{\top}(k))$ .

Bayesian estimation presents several advantages over the commonly used approach. Firstly, it combines the past information about the parameters with data in a natural and principled way. Particularly, when new observations are available, the previous posterior distribution is used as prior distribution integrating the past information in the new inference. Secondly, the Bayesian method handles missing data, outlier, multi-rate, multi-mode, bias update and noise simultaneously and optimally (Huang, 2011). Thirdly, it provides realistic and interpretable answers due to the ability to consider uncertainty in probability model. There are also disadvantages to using the Bayesian method. The method does not show how to select a prior distribution which requires skills to translate subjective prior beliefs into a mathematical form. The prior distribution may also heavily influence the posterior distribution generating misleading results. And, finally, the Bayesian method is associated with a higher computational cost.

### ■ Kalman filter

The Kalman filter is a parameter state estimator that uses indirect, inaccurate and uncertain observations. The term “filter” comes from the fact that the method finds the best estimate from noisy data to filtering out the noise. Although the Kalman filter is commonly used to estimate the values of the state vector of a dynamic system that is excited by stochastic disturbances and presents stochastic measurement noise, it may be applied to estimate model parameters. The state vector is augmented with the model parameters which are now denoted the augmentative states. The augmented state vector consisting of both the original state variables and the augmentative state variables is estimated by the Kalman filter. To set up an augmentative state variable, the behaviour of the augmentative states must be modelled (Halvorsen, 2014).

Consider a stationary stochastic vector signal  $\mathbf{x}(t)$  described according to Ikonen and Najim (2001) by

$$\mathbf{x}(t+1) = \mathbf{A}(t) \mathbf{x}(t) + \mathbf{B}(t) \mathbf{u}(t) + \mathbf{G}(t) \mathbf{v}(t) \quad (2.60)$$

$$\mathbf{y}(t) = \mathbf{C}(t) \mathbf{x}(t) + \mathbf{e}(t), \quad (2.61)$$

where  $\mathbf{x}(t)$  is the state vector,  $\mathbf{y}(t)$  is the vector of measurements,  $\mathbf{u}(t)$  is the vector of inputs,  $\mathbf{v}(t)$  is the system noise,  $\mathbf{e}(t)$  is the output noise,  $\mathbf{A}(t)$  is a system state transition matrix describing the internal dynamics of the system,  $\mathbf{B}(t)$  is the system input matrix,  $\mathbf{C}(t)$  is the output matrix describing the relation between states and measurements, and  $\mathbf{G}(t)$  is the noise transition matrix. The objective of the Kalman filter consists in the determination of the state vector  $\mathbf{x}(t)$  based on measurements  $\mathbf{y}(t)$  contaminated by noise  $\mathbf{e}(t)$ . Matrices  $\mathbf{A}$ ,  $\mathbf{B}$ ,  $\mathbf{C}$  and  $\mathbf{G}$  are assumed to be known and noises  $\mathbf{v}(t)$  and  $\mathbf{e}(t)$  are zero mean, independent Gaussian processes with known covariances  $\mathbf{V}(t)$  and  $\mathbf{Y}(t)$ .

In parameter estimation, it is supposed that the data is generated according to

$$\mathbf{y}(t) = \boldsymbol{\varphi}^\top(t) \boldsymbol{\theta} + \mathbf{e}(t), \quad (2.62)$$

where  $\boldsymbol{\varphi}(t)$  is the output matrix describing the relation between model parameters and measurements and  $\boldsymbol{\theta}$  are the model parameters. Supposing also that the prior distribution of  $\boldsymbol{\theta}$  is Gaussian with mean  $\boldsymbol{\theta}_0$  and covariance  $\mathbf{P}_0$ , the state-

space model defined by (2.60 and 2.61) may be rewritten as

$$\boldsymbol{\theta}(t+1) = \boldsymbol{\theta}(t) \quad (2.63)$$

$$\mathbf{y}(t) = \boldsymbol{\varphi}^\top(t) \boldsymbol{\theta}(t) + \mathbf{e}(t). \quad (2.64)$$

The Kalman filter algorithm is then applied by the following steps (Ikonen and Najim, 2001):

1. Initialize  $\hat{\mathbf{x}}(t_0|t_0)$  and  $P(t_0|t_0)$  for  $t_0$ .
2. Time update:  
Estimate the state estimate at  $t+1$  given data up to  $t$  by

$$\hat{\mathbf{x}}(t+1|t) = \mathbf{A}(t) \cdot \hat{\mathbf{x}}(t|t) + \mathbf{B}(t) \cdot \mathbf{u}(t). \quad (2.65)$$

Update the covariance matrix of the error in  $\hat{\mathbf{x}}(k+1|k)$  using

$$\mathbf{P}(t+1|t) = \mathbf{A}(t) \cdot \mathbf{P}(t|t) \cdot \mathbf{A}^\top(t) + \mathbf{G}(t) \cdot \mathbf{V}(t) \cdot \mathbf{G}^\top(t). \quad (2.66)$$

3. Measurement update:  
Observe the new measurements  $\mathbf{y}(t+1)$  at time  $tT$  (with  $T$  denoting the sampling time).  
Compute the Kalman filter gain matrix as

$$\mathbf{K}(t+1) = \mathbf{P}(t+1|t) \cdot \mathbf{C}^\top(t+1) \cdot \left[ \mathbf{Y}(t+1) + \mathbf{C}(t+1) \cdot \mathbf{P}(t+1|t) \cdot \mathbf{C}^\top(t+1) \right]^{-1}. \quad (2.67)$$

Correct the state estimate at  $t+1$  given data up to  $t+1$  with

$$\hat{\mathbf{x}}(t+1|t+1) = \hat{\mathbf{x}}(t+1|t) + \mathbf{K}(t+1) \cdot \left[ \mathbf{y}(t+1) - \mathbf{C}(t+1) \cdot \hat{\mathbf{x}}(t+1|t) \right]. \quad (2.68)$$

Update the new error covariance matrix

$$\begin{aligned} \mathbf{P}(t+1|t+1) = & \left[ \mathbf{I} - \mathbf{K}(t+1) \cdot \mathbf{C}(t+1) \right] \cdot \mathbf{P}(t+1|t) \cdot \left[ \mathbf{I} - \mathbf{K}(t+1) \cdot \mathbf{C}(t+1) \right]^\top + \\ & + \mathbf{K}(t+1) \cdot \mathbf{Y}(t+1) \cdot \mathbf{K}^\top(t+1). \end{aligned} \quad (2.69)$$



## ■ Other system identification approaches

### Principal Component Analysis

Principal Component Analysis (PCA) is a simple and non-parametric method for extracting important information (through principal components) from complex datasets. This technique provides a roadmap that drives to the reduction of a complex dataset to a lower dimension revealing simplified structures that often underlie it (Shlens, 2009; Jolliffe, 2002).

In order to assure equal importance of each variable on the model, PCA requires the normalization of the input data matrix  $X \in \mathbb{R}^{n \times m}$ . The normalization transforms the data to be zero mean and of unit variance as

$$X' = (X - \mathbf{1}_n \mathbf{b}^\top) \Sigma^{-1}, \quad (2.70)$$

where  $\mathbf{1}_n = [1 \ 1 \ \dots \ 1]^\top \in \mathbb{R}^{n \times 1}$ ,  $\Sigma = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_m)$  and  $\mathbf{b} = \frac{1}{n} X^\top \mathbf{1}_n$ . The variable  $\sigma_i$  is the standard deviation of each of the input variables. The normalized data,  $X' \in \mathbb{R}^{n \times m}$ , is then transformed to the score matrix,  $T \in \mathbb{R}^{n \times l}$ , of lower dimension using

$$X' = T P^\top + E, \quad (2.71)$$

where  $P \in \mathbb{R}^{m \times l}$  is the loading matrix and  $E \in \mathbb{R}^{n \times m}$  is the residuals matrix.

There are several ways to find the loading matrix. One of them is through the correlation matrix,  $C$ , which is calculated by

$$C = \frac{1}{n-1} (X')^\top X'. \quad (2.72)$$

Using this information, the eigenvalues  $\Lambda$  and the eigenvector matrix  $V$  of matrix  $C$  are derived from the decomposition

$$\Lambda = V^{-1} C V. \quad (2.73)$$

The diagonal eigenvalues of  $\Lambda$ ,  $\lambda_i$ , are sorted in descending order such that  $\lambda_1 > \lambda_2 > \dots > \lambda_m$ . The columns of  $P$  are formed by the eigenvectors corresponding

to the highest eigenvalues and

$$P = [V(\lambda_1) \ V(\lambda_2) \ \dots \ V(\lambda_l)], \quad (2.74)$$

where  $V(\lambda_i) \in \mathbb{R}^{m \times 1}$  is the vector from the  $V$  matrix corresponding to the eigenvalue  $\lambda_i$ .

After the determination of  $P$ , the eigenvalue decomposition in (2.71) must be performed to obtain  $T$  using approaches such as Singular Value Decomposition and NIPALS algorithm (Jolliffe, 2002; Kadlec et al., 2011).

At that point, it is possible to build a regression model via the least squares algorithm using the relation

$$\hat{\mathbf{y}} = T \boldsymbol{\theta}. \quad (2.75)$$

Considering the orthogonal property of  $T$ , the equation is simplified to

$$\boldsymbol{\theta} = (T^\top T)^{-1} T^\top \mathbf{y} = L^{-2} T^\top \mathbf{y}, \quad (2.76)$$

where  $L \in \mathbb{R}^{l \times l}$  is a diagonal matrix with elements equal to  $\sqrt{\Lambda_i}$  (Jolliffe, 2002).

Because the number of significant principal components may vary, it may be used an adaptive strategy to calculate these components may be used. There are several methods to calculate the number of principal components such as the cumulative percent variance, the scree test, the average eigenvalues, and the variance of reconstruction error (Valle et al., 1999; Li et al., 2000; Jolliffe, 2002; Jackson, 2003; Liu et al., 2009).

*Dynamic Principal Component Analysis:* For dynamic systems, the current values of the variables will depend on the past values. Thus, it is convenient to identify the linear relations between  $X(t)$  and  $X(t - l_a)$ , where  $l_a$  is the time-lag. The implementation of DPCA method consists in transforming the data matrix  $X(t)$  for a Hankel matrix (a set of repeated overlapping windows,  $X_H(l_a)$ ) and use it in the standard PCA method. The Hankel matrix transform is given by

$$X_H(l_a) = [X(t)X(t-1) \cdots X(t-l_a)]. \quad (2.77)$$

*Kernel Principal Component Analysis:* One of the main drawbacks of the PCA approach is its inability to model non-linear relationships between variables. Non-

linear PCA for the estimation of difficult-to-measure process variables may be achieved by applying non-linear relationships. Kernel PCA embeds the data into a high dimensional space (called feature space) performing a non-linear input transformation by the application of a non-linear function (called kernel function). Then, the PCA technique solves an eigenvalue problem to this new space without any non-linear optimization (Slišković et al., 2011).

Let  $\Phi$  be a mapping from the original space  $\mathbb{R}^m$  into a inner product space  $F$  so that

$$\Phi : \mathbb{R}^m \rightarrow F , \quad (2.78)$$

where the space  $F$  is referred to as the feature space.  $\Phi(X_i)$  represents the image of the data vector  $X_i \in \mathbb{R}^m$  in the feature space.

Let the kernel matrix  $K$  be a symmetric and positive semidefinite  $m \times m$  matrix with its elements defined by the inner product of all pairs of points  $\Phi(X_i)$  and  $\Phi(X_j)$  in the feature space so that

$$K_{ij} = \Phi(X_i) \cdot \Phi(X_j) , \quad i, j = 1, \dots, m . \quad (2.79)$$

There are a variety of kernel functions that can be used for kernel PCA. To introduce the kernel matrix  $K$  inside PCA algorithm, it has to be centered in the feature space as

$$\bar{\Phi}(X) = \Phi(X) - \frac{1}{m} \sum_{k=1}^m \Phi(X_k) \quad (2.80)$$

and the centered kernel matrix defined as

$$\bar{K} = \Phi(X) \cdot \Phi(X)^\top . \quad (2.81)$$

It is possible to derive the expression for centering the noncentered kernel matrix  $K$  by

$$\bar{K} = K - 1_m K - K 1_m + 1_m K 1_m , \quad (2.82)$$

where  $1_m$  is an  $m \times m$  matrix in which each element equals  $1/m$  (Schölkopf et al., 1998; Olsson, 2011). A simple algorithm for kernel PCA may be found in Fauvel et al. (2006).

### Partial Least Squares

Partial least squares (PLS) method is a dimensionality reduction technique that finds a reduced set of latent variables by maximizing the covariance between the process and quality spaces as explained in the reviews of Zhang and Zhang (2010) and Wold et al. (2001).

The method is able to deal with large dimensional collinear data as well as with the fact that the resulting model takes into account the covariance between input and output data. Because of this feature, prediction models of the difficult-to-measure variables based on PLS are more accurate than those based on the PCA method.

The aim of PLS is to project scaled and mean centered input data  $X' \in \mathbb{R}^{n \times m}$  and output data  $Y' \in \mathbb{R}^{n \times p}$  to separate latent variables:

$$X' = T P^T + E, \quad (2.83)$$

and

$$Y' = U Q^T + F, \quad (2.84)$$

where  $P^{m \times l}$  and  $Q^{p \times l}$  are the corresponding loading matrices,  $E$  and  $F$  are the input and output data residuals, respectively, and  $T \in \mathbb{R}^{n \times l}$  and  $U \in \mathbb{R}^{n \times l}$  are the score matrices or latent vectors

$$T = [\mathbf{t}_1 \ \mathbf{t}_2 \ \dots \ \mathbf{t}_l] \text{ with } \mathbf{t}_i \in \mathbb{R}^{n \times 1}, \quad (2.85)$$

and

$$U = [\mathbf{u}_1 \ \mathbf{u}_2 \ \dots \ \mathbf{u}_l] \text{ with } \mathbf{u}_i \in \mathbb{R}^{n \times 1}. \quad (2.86)$$

The latent vectors (orthogonal to each other:  $\mathbf{t}_i \mathbf{u}_i^T = 0, \forall i \neq j$ ) represent a more compact description of the input data achieved by removing the collinearity from the data. The columns  $\mathbf{p}_i \in \mathbb{R}^m$  and  $\mathbf{q}_i \in \mathbb{R}^p$  of the loading matrices  $P$  and  $Q$  represent the contributions of the input and output variables to the latent vectors  $\mathbf{t}$  and  $\mathbf{u}$ , respectively.

The PLS method produces a regression model between the latent scores defined as

$$U = T B + R, \quad (2.87)$$

where  $B \in \mathbb{R}^{l \times l}$  is the diagonal matrix of the regression weights that is calculated by minimizing the regression residuals  $R$ . The estimated variables  $\hat{Y}$  are given by

$$\hat{Y} = T B Q^T. \quad (2.88)$$

There are several forms to calculate the vectors  $\mathbf{t}$ ,  $\mathbf{p}$ ,  $\mathbf{u}$ ,  $\mathbf{q}$  and  $\mathbf{b}$ . For instance, NIPALS algorithm defines these vectors as

$$\mathbf{t}_i = \frac{E_{i-1} \mathbf{w}_i}{\|E_{i-1} \mathbf{w}_i\|}, \quad (2.89)$$

$$\mathbf{p}_i = E_{i-1}^T \mathbf{t}_i, \quad (2.90)$$

$$\mathbf{u}_i = F_{i-1} \mathbf{q}_i, \quad (2.91)$$

$$\mathbf{q}_i = \frac{F_{i-1}^T \mathbf{t}_i}{\|F_{i-1}^T \mathbf{t}_i\|}, \quad (2.92)$$

$$\mathbf{b}_i = u_i^T \mathbf{t}_i, \quad (2.93)$$

where  $i$  is the index of the latent variable and

$$\mathbf{u}_i = F_{i-1} \mathbf{q}_i. \quad (2.94)$$

After each iteration the residuals are deflated

$$E_{i+1} = E_i - \mathbf{t}_i \mathbf{p}_i^T, \quad (2.95)$$

$$F_{i+1} = F_i - \mathbf{u}_i \mathbf{q}_i^T, \quad (2.96)$$

which is followed by the calculation of the next  $(i + 1)^{\text{th}}$  vectors for PLS models using the new data matrices  $E_{i+1}$  and  $F_{i+1}$ . The number of calculated latent dimensions is usually established using cross-validation or some other parameter optimization technique.

## Support Vector Machines

Support Vector Machines (SVM) is a machine learning method for learning linear and non-linear rules. Joachims (2005) presents a method review explaining the fundamentals and the aspects of its implementation.

The method is based on statistical learning theory and has gained more attention in the field of soft sensors.

Its first step is the construction of a linear function in a high-dimensional space,  $m_k$ , i.e.,  $\varphi() := \mathbb{R}^m \rightarrow \mathbb{R}^{m_k}$  with the objective of optimizing the parameters  $\omega$  and  $b$  in order to fulfil the condition

$$|y - \omega^T \varphi(\mathbf{x}) - b| < \epsilon, \quad (2.97)$$

where  $\epsilon$  is the precision parameter.

After some mathematical modifications, the representation of the model is expressed by

$$\hat{y} = \sum_i (\alpha_i - \alpha_i^*) k(\mathbf{x}_i, \mathbf{x}) + b, \quad (2.98)$$

where  $\mathbf{x}_i$  are the support vectors,  $k()$  may be any function fulfilling the Mercer condition<sup>2</sup>, and  $b$  is a constant that may be calculated by applying the Karush-Kuhn-Tucker conditions.

In order to determine  $\mathbf{x}_i$ ,  $\alpha_i$ ,  $\alpha_i^*$  and  $b$ , the method uses quadratic programming techniques. Although the complexity of the determination problem is independent of the dimensionality of the input space, it grows with the number of training samples and can even become computationally infeasible.

## Subspace identification

Subspace identification has received a lot of attention in the recent years (van Overschee and de Moor, 1996; Qin, 2006; Doraiswami and Cheded, 2014). This fact is essentially due to its numerical efficiency and robustness, as well as to the minimal requirement of a priori information (such as the structure of the system). The only design parameter is the threshold value for the singular values truncation. The subspace identification does not require non-linear optimization to

---

<sup>2</sup>Mercer condition:  $k(\mathbf{x}_i, \mathbf{x}) = \varphi(\mathbf{x}_i)^T \varphi(\mathbf{x}_j)$ .

calculate the model parameters, is based on tools computationally reliable (such as the singular value decomposition), is non-recursive, and avoids problems associated with optimization and possible local minima (Doraiswami and Cheded, 2014; van Overschee and de Moor, 1996). Such advantages have made it an appealing technique.

The subspace identification may be categorized into two classes: the open- and the closed-loop identification. The open-loop identification, where the input data is assumed to be independent of the past noise, is the most popular class and includes the N4SID (Numerical algorithms for Subspace State Space System IDentification), the MOESP (Multivariable Output Error State Space) and the CVA (Canonical Variate Analysis) methods.

Given a set of input and output measurements and the state-space model of combined deterministic and stochastic system in an innovation form

$$x_{k+1} = Ax_k + Bu_k + Ke_k, \quad (2.99)$$

$$y_k = Cx_k + Du_k + e_k, \quad (2.100)$$

where  $k$  is an arbitrary time index,  $u_k \in \mathbb{R}^{n_u}$  is the input,  $x_k \in \mathbb{R}^n$  is the state,  $y_k \in \mathbb{R}^{n_y}$  is the output,  $K \in \mathbb{R}^{n \times n_y}$  is the steady-state Kalman gain, and  $e_k \in \mathbb{R}^{n_y}$  is an unknown innovation with covariance matrix  $R = E[e_k e_k^\top]$  (Trnka, 2005, Section 3.1). The subspace identification problem aims to estimate the system order  $n$  and obtains system matrices  $A \in \mathbb{R}^{n \times n}$ ,  $B \in \mathbb{R}^{n \times n_u}$ ,  $C \in \mathbb{R}^{n_y \times n}$  and  $D \in \mathbb{R}^{n_y \times n_u}$ , the gain matrix  $K \in \mathbb{R}^{n \times n_y}$ , and the noise covariance matrix  $R \in \mathbb{R}^{n_y \times n_y}$ , as explained by Qin (2006).

Based on the innovation form, an extended model may be formulated as

$$Y_f = \Gamma_f X_f + H_f U_f + G_f E_f, \quad (2.101)$$

where the uppercase denotes Hankel matrices. Subscripts  $p$  and  $f$  are used to denote past and future values. The output Hankel matrices  $Y_p$  and  $Y_f$  are composed

by

$$Y_p = \begin{bmatrix} y_0 & y_1 & \cdots & y_{n_j-1} \\ y_1 & y_2 & \cdots & y_{n_j} \\ \vdots & \vdots & \ddots & \vdots \\ y_{n_i-1} & y_{n_i} & \cdots & y_{n_i+n_j-2} \end{bmatrix} \in \mathbb{R}^{n_i n_y \times n_j}, \quad (2.102)$$

and

$$Y_f = \begin{bmatrix} y_{n_i} & y_{n_i+1} & \cdots & y_{n_i+n_j-1} \\ y_{n_i+1} & y_{n_i+2} & \cdots & y_{n_i+n_j} \\ \vdots & \vdots & \ddots & \vdots \\ y_{n_i+n_h-1} & y_{n_i+n_h} & \cdots & y_{n_i+n_h+n_j-2} \end{bmatrix} \in \mathbb{R}^{n_h n_y \times n_j}, \quad (2.103)$$

where  $n_i$  is the number of rows of the Hankel matrix composed by past values,  $n_h$  is the number of rows of the Hankel matrix composed by future values, and  $n_j$  is the number of columns of each of those matrices. Similar forms are applied for  $U$  and  $E$ . The extended observability matrix  $\Gamma_f \in \mathbb{R}^{n_h n_y \times n_h n}$  and the Toeplitz matrices  $H_f \in \mathbb{R}^{n_h n_y \times n_h n_u}$  and  $G_f \in \mathbb{R}^{n_h n_y \times n_h n_y}$  are defined by

$$\Gamma_f = \begin{bmatrix} C \\ CA_K^1 \\ \vdots \\ CA_K^{n_h-1} \end{bmatrix}, \quad H_f = \begin{bmatrix} D & 0 & \cdots & 0 \\ CB & D & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ CA^{n_h-2}B & CA^{n_h-3}B & \cdots & D \end{bmatrix}, \quad (2.104)$$

and

$$G_f = \begin{bmatrix} I & 0 & \cdots & 0 \\ CK & I & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ CA^{n_h-2}K & CA^{n_h-3}K & \cdots & I \end{bmatrix}, \quad (2.105)$$

where  $A_K = [A - KC]$ . The Kalman state sequences  $X_k$  are estimated using

$$X_k = L_p Z_p + A_K^p X_{k-n_i}, \quad (2.106)$$

where  $L_p$  is the predictor controllability matrix,  $Z_p = [U_p, Y_p]^\top$  and  $X_{k-n_i} = [x_{k-n_i}, x_{k-n_i+1}, \cdots, x_{k-n_i+n_j-1}]$ . Matrix  $A_K^p$  is approximately zero matrix for a suf-



ficiently large value of  $n_i$ . Together, (2.101) and (2.106) become

$$Y_f = H_{fp}Z_p + H_fU_f + G_fE_f, \quad (2.107)$$

where  $H_{fp} = \Gamma_f L_p$ . Under open-loop conditions,  $E_f$  is uncorrelated to  $U_f$  and to  $Z_p$  which means that

$$\lim_{N \rightarrow \infty} \frac{1}{N} E_f U_f^\top \rightarrow 0,$$

and

$$\lim_{N \rightarrow \infty} \frac{1}{N} E_f Z_p^\top \rightarrow 0.$$

The open-loop subspace identification algorithms involve the following steps: projection or regression, model reduction, and parameter estimation (Qin, 2006):

### 1. Projection or regression step:

In this step, the elimination of the input and noise terms in (2.107) is done. To eliminate the input term, the equation is orthogonally projected onto the row space orthogonal to  $U_f$  through

$$\begin{aligned} Y_f \Pi_{U_f}^\perp &= H_{fp} Z_p \Pi_{U_f}^\perp + H_f U_f \Pi_{U_f}^\perp + G_f E_f \Pi_{U_f}^\perp \\ &= H_{fp} Z_p \Pi_{U_f}^\perp + G_f E_f, \end{aligned} \quad (2.108)$$

where  $\Pi_{U_f}^\perp$  is the projection matrix to the orthogonal complement of  $U_f$ . Two properties of  $\Pi_{U_f}^\perp$  are used in this derivation:  $U_f \Pi_{U_f}^\perp = 0$  and  $E_f \Pi_{U_f}^\perp = E_f$  (Qin, 2006). The noise term  $G_f E_f$  is left intact because it is uncorrelated with the deterministic input  $U_f$ .

To estimate  $H_{fp}$ , a least squares step is performed such as

$$\hat{H}_{fp} = \arg \min_{H_{fp}} \|Y_f \Pi_{U_f}^\perp - H_{fp} Z_p \Pi_{U_f}^\perp\|_F^2. \quad (2.109)$$

### 2. Model reduction step:

This step consists in the reduction of the model order to an appropriate observable low dimensional subspace. Firstly, it is applied a singular value decomposition on

$$W_1 \hat{H}_{fp} W_2 = USV^\top, \quad (2.110)$$

where according to Qin (2006)

$$\begin{array}{ll}
 \text{for regression approach:} & W_1 = I \qquad W_2 = I \\
 \text{for N4SID:} & W_1 = I \qquad W_2 = (Z_p Z_p^\top)^{1/2} \\
 \text{for MOESP:} & W_1 = I \qquad W_2 = (Z_p \Pi_{U_f}^\perp Z_p^\top)^{1/2} \\
 \text{for CVA:} & W_1 = (Y_f \Pi_{U_f}^\perp Y_f^\top)^{1/2} \quad W_2 = (Z_p \Pi_{U_f}^\perp Z_p^\top)^{1/2}.
 \end{array}$$

The system order  $n$  is then determined by inspecting the singular values  $S$  and matrices from the definition of  $H_{fp}$  are reduced to  $\Gamma_f = W_1^{-1} U_n S_n^{1/2}$  (Qin, 2006, Section 3.1.4) and  $L_p = S_n^{1/2} V_n^\top W_2^{-1}$  (Qin, 2006, Section 4.2).

3. **Parameter estimation:** Matrices  $A$  and  $C$  are calculated from the extended observability matrix  $\Gamma_f$ . Matrix  $C$  is directly read from the  $n_y$ -dimension first block row of  $\Gamma_f$  and matrix  $A$  is determined from the shift structure defined as

$$\Gamma_f^{(1)} A = \Gamma_f^{(2)}, \quad (2.111)$$

where  $\Gamma_f^{(1)}$  and  $\Gamma_f^{(2)}$  are the matrix  $\Gamma_f$  without the last block row and without the first block row, respectively. This equation is linear and may be solved by the least squares method to extract  $A$ .

Subsequently, matrices  $B$  and  $D$  are computed. Considering the purely deterministic case,  $G_f E_f = 0$ , multiplying (2.101) by the projection matrix to the orthogonal complement of  $\Gamma_f$  ( $\Pi_{\Gamma_f}^\perp$ ) and by the transpose of  $U_f$ , in such way that  $\Pi_{\Gamma_f}^\perp \Gamma_f = 0$  and  $U_f U_f^\top = I$ , one obtains

$$\Pi_{\Gamma_f}^\perp Y_f U_f^\top = \Pi_{\Gamma_f}^\perp H_f. \quad (2.112)$$

The equation is now rewritten as

$$\begin{aligned}
 (\mathcal{M}_1 \ \mathcal{M}_2 \ \mathcal{M}_3 \ \cdots \ \mathcal{M}_i) &= \\
 &= (\mathcal{L}_1 \ \mathcal{L}_2 \ \mathcal{L}_3 \ \cdots \ \mathcal{L}_i) \begin{pmatrix} D & 0 & 0 & \cdots & 0 \\ CB & D & 0 & \cdots & 0 \\ CAB & CB & D & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ CA^{i-2}B & CA^{i-3}B & CA^{i-4}B & \cdots & D \end{pmatrix}.
 \end{aligned} \tag{2.113}$$

where  $\mathcal{M}$  is the left hand side of the equation and  $\mathcal{L}$  the projection matrix  $\Pi_{\Gamma_f}^\perp$ , for simplicity of notation. Because this equation is linear in  $B$  and  $D$ , it is reformulated as

$$\begin{pmatrix} \mathcal{M}_1 \\ \mathcal{M}_2 \\ \mathcal{M}_3 \\ \vdots \\ \mathcal{M}_i \end{pmatrix} = \begin{pmatrix} \mathcal{L}_1 & \mathcal{L}_2 & \cdots & \mathcal{L}_{i-1} & \mathcal{L}_i \\ \mathcal{L}_2 & \mathcal{L}_3 & \cdots & \mathcal{L}_i & 0 \\ \mathcal{L}_3 & \mathcal{L}_4 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathcal{L}_i & 0 & \cdots & 0 & 0 \end{pmatrix} \begin{pmatrix} I_l & 0 \\ 0 & \Gamma_i^{(2)} \end{pmatrix} \begin{pmatrix} D \\ B \end{pmatrix}, \tag{2.114}$$

which is typically overdetermined and may be solved using least squares (Trnka, 2005; van Overschee and de Moor, 1996, Section 2.4.2).

Finally, stochastic matrices  $K$  and  $R$  are determined from the covariance estimate of the residuals as

$$\begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix} = \frac{1}{j - (n + m)} \epsilon \epsilon^\top, \tag{2.115}$$

$$R = \Sigma_{22}, \tag{2.116}$$

$$K = \Sigma_{12} \Sigma_{22}^{-1}. \tag{2.117}$$

### ■ Time delay estimation

The time delay estimation is an important problem and, therefore, has received much attention by the research community. Most of the proposed methods may be classified into four major classes as schematized in Figure 2.9: the time-delay

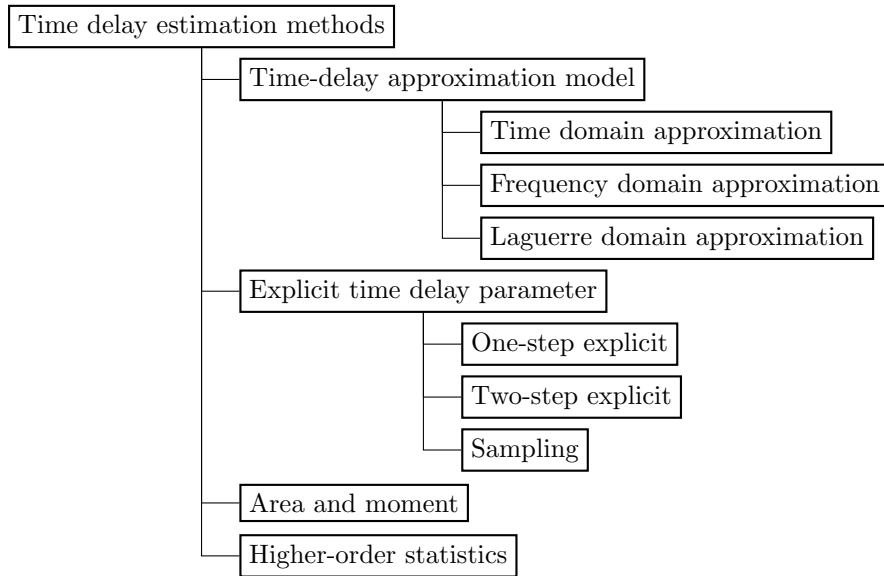


Figure 2.9: Classification of time delay estimation methods.

approximation model methods, the explicit time delay parameter methods, the area and moment methods, and the higher-order statistics methods (Björklund, 2003; Björklund and Ljung, 2003).

The distinction between the first two classes is the usage of the time delay parameter as an explicit parameter in the model: time delay approximation model methods use a model relating the input and output signals not containing explicitly the time delay parameter while the class of explicit time delay parameter methods does not. Essentially, the methods of the first considered class estimate the model and only then estimate the time delay from the already estimated model. These class contains three sub-classes: (1) the time domain approximation methods where, from an impulse response, the time delay is seen as the delay observed in the response to the impulse stimulus for the impulse to start and may be calculated finding the peak of the cross-correlation between the input and output signals; (2) the frequency domain approximation methods where the time delay is estimated from the phase of the time delay  $e^{-iw\theta}$ ; and (3) the Laguerre domain approximation methods where the time delay is estimated from a relation between the input and the output signals expressed in basis functions such as Laguerre or Kautz functions.

In what concerns the explicit time delay parameter methods, there are three

main sub-classes: (1) the one-step explicit methods where the model parameters and the time delay are calculated simultaneously; (2) the two-step explicit methods where the model parameters and the time delay are calculated sequentially; and (3) the sampling methods where the sampling process is utilized to derive an expression for the time delay.

The area and moment methods use relations between the time delay and certain areas over or below the step response, or certain moments of the impulse response. Basically, the methods independently perform the estimation of the step or impulse response and estimate the time delay from those responses.

Finally, the higher order statistics methods use, as the class name suggests, higher order statistics, such as the bi-spectrum and 3<sup>rd</sup> order moments, to estimate the time delay.

### 2.2.5 Model validation

As Ljung (1999) and Lyzell (2009) refer, model validation is the phase that ensures the validity of the developed model through the evaluation of its performance. Indeed, it is not recommended to compare only visually the profiles of the predicted and the measured outputs drawn side by side (Ye, 2003). Several complementary methods have been developed to evaluate the model performance determining, usually, its accuracy and reliability. Accuracy is associated to the agreement level between predicted and the real outputs, while reliability is the variation degree of the prediction errors (Khatibisepehr et al., 2013).

Most of the model validation methods are based on the residuals

$$\epsilon(t, \hat{\theta}) = y(t) - y(t, \hat{\theta}), \quad (2.118)$$

where  $\hat{\theta}$  is the parameter estimate. Cross-validation and cross-correlation methods are examples of these methods. Both validate the model using the output predicted from a new dataset (different from the dataset used to estimate the parameters) for comparing to the measured output. Different criteria were proposed to quantify the closeness of the predicted and the measured outputs.

Cross-validation method usually employs the model fit value given by

$$\text{fit} = 100 \left( 1 - \frac{\|\epsilon(t, \hat{\theta})\|_2}{\|y(t) - \bar{y}\|_2} \right) \quad (2.119)$$

as its criterion. The model fit value provides the relative performance increase of using the developed model compared to using the output mean  $\bar{y}$  as a predictor. However, this criterion depends on the amount of noise in the data (more noise lowers the model fit). The cross-validation method is important because it prevents the over-fitting of the identification data (Lyzell, 2009).

As an alternative method is the cross-correlation method which is based on the covariance between the residuals and the input variable  $u(t)$  described by

$$r_{eu} = \frac{1}{N} \sum_{t=1}^N \epsilon(t) u(t - \tau), \quad (2.120)$$

where  $\tau$  is the time shift and  $N$  is the number of samples contained in the dataset. If the developed model is capable of describing the dynamic system, the residuals follows a Gaussian distribution and are independent of the input signal. Therefore, if the residuals have these characteristics, the cross-correlation is zero for all  $\tau$  values. But, if they correlate with the input variable, the cross-correlation is non-zero, suggesting that the model does not capture the system. Auto-correlation method may also be informative to consider the correlation among the residuals themselves. Large values indicate that the predicted output could have been predicted from past data and, consequently, better (Ye, 2003; National Instruments Corporation, 2015).

Cross-validation and cross-correlation methods proved to be quite effective in practice although they are very simple. However, they use different metrics that focus on detecting one particular deficiency of the model under validation. Therefore, it is important to use other criteria that help to overcome other deficiencies.

Among them, the mean squared error (MSE) and the mean absolute deviation

(MAD) are very well known in practice (Ye, 2003). These measures are defined as

$$\text{MSE} = \frac{1}{N} \sum_{t=1}^N \epsilon(t, \hat{\theta})^2, \quad (2.121)$$

$$\text{MAD} = \frac{1}{N} \sum_{t=1}^N |\epsilon(t, \hat{\theta})|. \quad (2.122)$$

While MSE amplifies larger errors, the MAD simply takes the absolute values of the error.

Sotomayor et al. (2003) proposed two new measures for model validation: the mean relative squared error (MRSE) and the mean variance-accounted-for (MVAF) that are defined by

$$\text{MRSE} = \frac{1}{l} \sum_{i=1}^l \sqrt{\frac{\sum_{t=1}^N \epsilon_i(t, \hat{\theta})^2}{\sum_{t=1}^N y_i(t)^2}} \times 100, \quad (2.123)$$

$$\text{MVAF} = \frac{1}{l} \sum_{i=1}^l \sqrt{1 - \frac{\sigma^2(\epsilon_i(t, \hat{\theta}))}{\sigma^2(y_i(t))}} \times 100, \quad (2.124)$$

where  $l$  is the number of output variables and  $\sigma^2$  is the variance. Similarly to the MSE index, the MRSE index allows to measure a relative error with the zero value indicating a perfect model. MVAF index evaluates the dynamic properties of the developed models and, when the index is close to 1, it indicates a model reproducing well the dynamic properties of the real system.

The goodness of the fit is a metric that takes advantage of the datasets used for model development and validation. This metric, obtained through the ratio

$$G = \frac{\sum_{t \in \{\text{validation}\}} \epsilon(t, \hat{\theta})}{\sum_{t \in \{\text{development}\}} \epsilon(t, \hat{\theta})}, \quad (2.125)$$

is almost always larger than unity because the model development dataset was used to estimate the parameters and, therefore, presents less error (Good, 2006, Section 8.5.1).

In addition to the evaluation of the predictive capabilities of the model, the model validation may also include the evaluation of the complexity of the model, particularly the higher order structures, such as the Akaike information crite-

ria (AIC, Akaike (1969)) and the minimum description length (MDL, Rissanen (1978)). Mathematically, they are defined by

$$\text{AIC} = \left( 1 + \frac{2 \dim(\hat{\theta})}{N} \right) \cdot \frac{1}{N} \sum_{t=1}^N \epsilon(t, \hat{\theta})^2, \quad (2.126)$$

$$\text{MDL} = \left( 1 + \frac{2 \dim(\hat{\theta}) \log(N)}{N} \right) \cdot \frac{1}{N} \sum_{t=1}^N \epsilon(t, \hat{\theta})^2, \quad (2.127)$$

where  $\dim$  operator is the number of non-zero elements of a vector (Lyzell, 2009, Section 2.4).

When dealing with categorical outputs (usually denominated as classification), other specific measures are used to estimate the error. Some examples are the average Kullback-Leibler distance, false positives and negatives, precision, recall and  $F$  measure, sensitivity and specificity, receiver operating characteristic and lift curves (Ye, 2003).

Usually, the training methods assume that the training data is representative. However, this assumption may not fully hold and, consequently, it is important to separate training from testing data to estimate independently the model error. When the dataset is sufficiently large, the available data is decomposed into three subsets: the training, the validation, and the test datasets. While the training data is used to develop the model, the validation data is used to re-tune the developed model, and the test set is used to evaluate the prediction performance. When the dataset is relatively small, data usually is re-sampled (Khatibisepehr et al., 2013). The  $k$ -fold cross-validation, the leave-one-out cross-validation, and the bootstrap re-sampling are useful techniques that re-sample the data (Ye, 2003; Lahiri, 2003). The  $k$ -fold cross-validation divides the data into  $k$  partitions (usually  $k = 10$ ), applies the identification method to  $k - 1$  partitions (for  $k = 10$ , 90% of the data) and uses the remaining partition for testing. If all the  $n$  points of the dataset are used to estimate the performance, the leave-one-out cross-validation consisting of a special case of the  $k$ -fold cross-validation for  $k = N$  is used. A more efficient method is the bootstrap re-sampling that generates a dataset for the identification by randomly sampling with replacement. This data point selection approach does not use a fraction of the data in the identification phase, reserving those



points for validation purposes.

## **2.3 Using first- and second-order state-space models for system identification<sup>3</sup>**

System identification plays an important role in the development of process simulators and controllers. The ability to determine correctly the model parameters directly affects the model quality and, therefore, the model based controller performance. This section presents a detailed case study of the development of a system identification approach and its computational implementation based on sequential quadratic programming (SQP) in which first and second-order linear systems, represented in state-space, are identified from simulated and from real industrial process data. Both single-input single-output and multivariable processes are considered.

### **2.3.1 Parameter estimation based on numerical optimization**

The first step of system identification consists of the so-called *process activation*. During this procedure the process is subjected to a set of disturbances whose magnitude should be carefully chosen. Indeed, if the process is activated too aggressively, the disturbance may impact the product quality and even the process safety. On the other hand, if the activation is not enough, an accurate process model cannot be obtained because the information content of the activated dataset is too low and the uncertainties (due for example to measurement noise and other disturbances) may become dominant (Sung et al., 2009; Ljung, 1999). Once process data with sufficient information is collected, the model parameters are determined such that the model response reproduces the observed response of the actual process.

The most frequently used curve fitting criterion is the least squares criterion which penalises the standard deviation of the model predictions from the dataset.

---

<sup>3</sup>This section is a reproduction in part from Brásio, A. S., Romanenko, A., and Fernandes, N. C. (2015c). Using sequential quadratic programming for system identification. *Applied Mathematics & Information Sciences*, 9(1):19–26. URL <http://www.naturalspublishing.com/Article.asp?ArtcID=7413>. Copyright 2015 Natural Sciences Publishing.

Another common criterion is the sum of the absolute deviation. However, the latter is not continuous and that poses additional challenges in the optimization problem. The Chebyshev approximation criterion minimizes the largest absolute deviation over the entire set. However, this criterion is often difficult to apply in practice since the resulting optimization problem may require advanced mathematical procedures (Leon, 2012).

The nonlinear constrained optimization problem is defined as

$$\underset{\mathbf{p}}{\text{minimize}} \quad J(\mathbf{y}, \mathbf{u}, \mathbf{p}) \quad (2.128a)$$

$$\text{subject to} \quad \dot{\mathbf{y}} = f(\mathbf{y}, \mathbf{u}, \mathbf{p}) \quad (2.128b)$$

$$\mathbf{y}_L \leq \mathbf{y} \leq \mathbf{y}_U \quad (2.128c)$$

$$\mathbf{u}_L \leq \mathbf{u} \leq \mathbf{u}_U \quad (2.128d)$$

$$\mathbf{p}_L \leq \mathbf{p} \leq \mathbf{p}_U \quad (2.128e)$$

$$g(\mathbf{p}) \leq 0, \quad (2.128f)$$

where  $J$  denotes the objective function,  $\mathbf{p}$  is the model parameters vector to be estimated,  $\mathbf{x}$  and  $\mathbf{u}$  are the vectors of state and input variables (respectively), and the subscripts  $_L$  and  $_U$  stand for lower and upper bounds (respectively). The set of equations (2.128b) defines a set of constraints arising from the model dynamics. Inequalities (2.128f) may enforce additional identification criteria.

Given a model  $\mathbf{y} = f(\mathbf{y}, \mathbf{u}, \mathbf{p}) \in \mathbb{R}^{n_y}$  and a set of  $m \times n_y$  data points  $(t_i, \mathbf{y}_{\text{exp},i})$ , the objective function  $J$  is written, according to the minimum least squares criterion, as

$$J = \sum_{i=1}^m [\mathbf{y}_{\text{exp},i} - \mathbf{y}_i]^\top \mathbf{Q} [\mathbf{y}_{\text{exp},i} - \mathbf{y}_i], \quad (2.129)$$

where  $\mathbf{Q}$  is a diagonal matrix containing the weights given to each observed variable. In this work, equal weight was given to all output variables and thus  $\mathbf{Q}$  is the  $n_y \times n_y$  identity matrix.

It should be noted that generally (2.128) may become nonconvex causing numerical difficulties and local minima. However, since in this work the parameters belong to a linearized model the number of the decision variables is low, the Sequential Quadratic Programming (SQP) exhibited satisfactory performance. Further solution refinement may be achieved via multistarting (György and Kocsis,

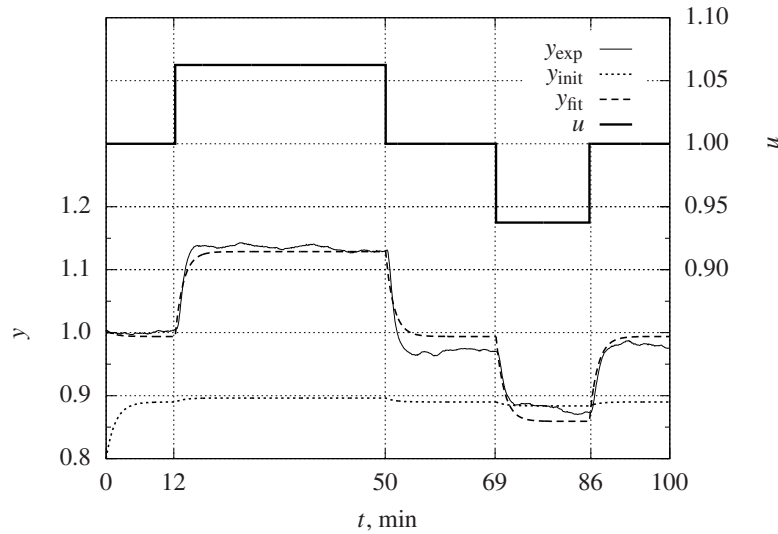


Figure 2.10: SISO system identification using an FO model.

2011).

## 2.3.2 Discussion of results

### ■ SISO systems identification

An industrial heat exchanger installed in a process plant, which may be regarded as a SISO system, was stimulated with a sequence of input steps and the profiles of the input and the output variables were registered. The obtained dataset contains 1200 points covering an interval of 100 minutes with a sampling period of 5 seconds. For confidentiality reasons the data was later normalized.

Both the stimuli,  $u$ , and the system response,  $y_{\text{exp}}$ , obtained during the process activation stage may be seen in Figure 2.10 (as well as in Figure 2.11). The success of system identification strongly depends on the quality of the data and, therefore, on its signal to noise ratio (SNR). The collected industrial dataset is characterized by an SNR of 11.0.

The optimization procedure described above was used to identify the system. The implementation was made in GNU Octave 3.6.3 using its general nonlinear minimization via `sqp()` sequential quadratic programming solver. Based on the shape of the experimental response curve, both FO and SO models were tested (see (2.24) and (2.27)). The set of optimization related conditions and the obtained

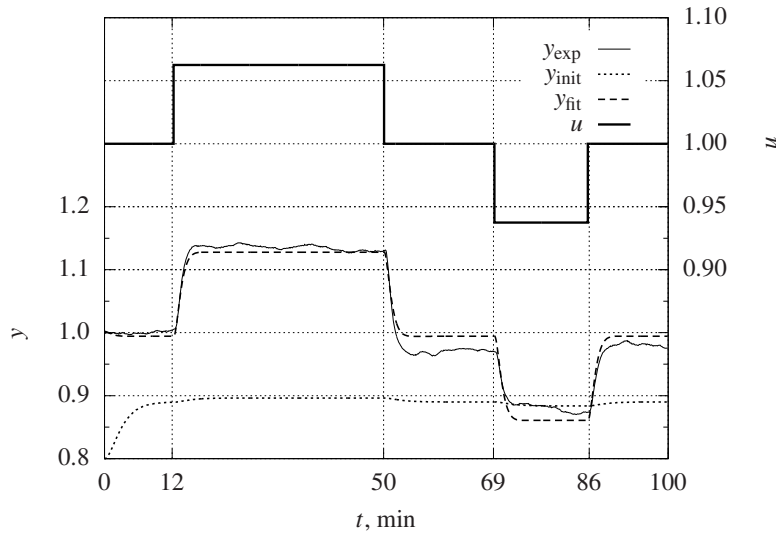


Figure 2.11: SISO system identification using an SO model.

Table 2.1: Identification results for the SISO system using both FO and SO models.

<b>p</b>	Initial	LB	UB	Fit	Indicators
<i>FO model</i>					
$K_p$	0.100	0.0001	5	2.154	
$\tau$	100.000	1	1000	88.868	$J = 0.286$
$\bar{x}$	0.800	-10	10	1.004	$R^2 = 0.9727$
$\bar{u}$	0.100	-10	10	1.005	
<i>SO model</i>					
$K_p$	0.100	0.0001	5	2.133	
$\omega$	0.010	0	1	0.022	$J = 0.263$
$\xi$	1.000	0.0001	10	0.919	$R^2 = 0.9749$
$\bar{x}$	0.800	-10	10	1.002	
$\bar{u}$	0.100	-10	10	1.004	

Dataset has SNR=11.0

model parameters as well as some fitting quality indicators are presented in Table 2.1.

The stopping criterion of the `sqp()` solver was set to  $10^{-6}$  in both cases (FO and SO). The dynamic responses of the mentioned models are drawn in Figures 2.10 and 2.11 (dashed line) for comparison with the real system response (thin solid line).

### 2.3. Using first- and second-order state-space models for system identification

---

It is noteworthy that the SNR of the data is relatively significant and that the initial guess for the parameters is poor (as it is shown by the dotted line representing the model prediction with the first iteration parameters). Although these two factors make the identification process more difficult, both FO and SO resulting models are able to capture well the process dynamics, as proven by the high correlation factors,  $R^2$ .

Both models present a comparable performance, attested by similar values of the objective function and also by similar values of  $R^2$  (see Table 2.1). By comparison of Figures 2.10 and 2.11, it is possible to conclude that the predictions of both models are, in this case, quite similar.

Therefore, and in this specific situation, one should select the FO model since it is able to achieve the same performance as the SO model but with a simpler structure. The lower number of parameters of the FO model also reduces the computational effort required in the fitting.

#### ■ MIMO systems identification

A continuous stirred tank reactor (CSTR) equipped with a heating coil is a good example of a MIMO system commonly used in industry. This system has two input variables (the inlet flow concentration of reactant A,  $C_{A,i}$ , and the temperature of the heating fluid in the coil,  $T_c$ ) and two output variables (the concentration of reactant in the reactor,  $C_A$ , and the temperature in the reactor,  $T$ ).

In order to collect data for the identification of a CSTR subjected to external heating, a simulation run was carried out using the first principles model (Appendix A). The timespan of the data is 1000 minutes with a sampling interval of 1 min. This dataset exhibits SNR of 8.6 and 3.8 for  $T_{\text{exp}}$  and  $C_{A,\text{exp}}$ , respectively.

The input used to stimulate the system and the generated experimental results are plotted in Figure 2.12. The interaction among the variables is clear: for instance, a disturbance in input variable  $T_c$  results in a dynamic response not only of  $T$  but also of the second output variable,  $C_A$ . Similarly, by activating the input variable  $C_{A,i}$  both output variables are affected.

First-order model:

Using an FO model whose state variables vector coincide with the output variables vector  $\mathbf{x}(t) = \mathbf{y}(t) = \begin{bmatrix} T & C_A \end{bmatrix}^\top$ , all  $\mathbf{A}$ ,  $\mathbf{B}$ , and  $\mathbf{C}$  matrices have

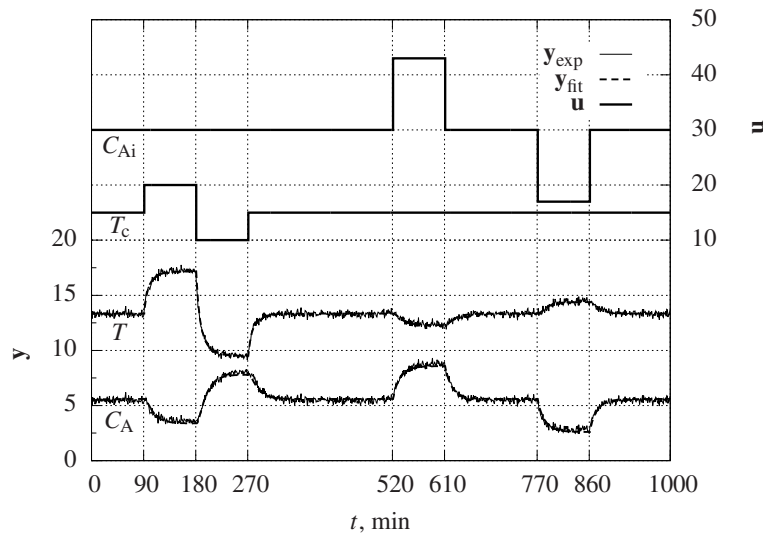


Figure 2.12: MIMO system identification using an FO model.

dimension  $2 \times 2$  and matrix  $C$  is the identity matrix. Also, from a priori physical/chemical analysis of the system, it is possible to conclude that the first input variable ( $T_c$ ) has a direct effect on  $T$  while it has an indirect effect on  $C_A$  through the variable  $T$ . Moreover, that the effect of the second input variable ( $C_{A,i}$ ) is direct on  $C_A$  but indirect on  $T$ . These facts may be used to reduce to 10 the number of parameters to be estimated through optimization for the FO model, since  $B_{12} = B_{21} = 0$ .

The parameter values of this system determined by the `sqp()` solver are summarized in Table 2.2.

In spite of the high level of noise, especially in the second variable ( $\text{SNR} = [8.6 \ 3.8]$ ), the obtained correlation factor was even higher than in the case of the SISO system, revealing an excellent fit quality. The model response with the optimized parameters is drawn (dashed line) in Figure 2.12 together with the experimental response of the system (thin solid line) for easy comparison. The model is able to capture the peculiarities of the system, namely the strong interactions among its variables.

Second-order model:

The identification of the MIMO system is also carried out via an SO approach. The state variables vector was defined as  $\mathbf{x}(t) = [T \ C_A \ \dot{T} \ \dot{C}_A]^T$

### 2.3. Using first- and second-order state-space models for system identification

Table 2.2: Identification results for the MIMO system using FO and SO models.

$\mathbf{p}$	Initial	LB	UB	Fit	Indicators
<i>FO model</i>					
$\mathbf{A}_{11}$	$-1 \cdot 10^{-2}$	-1	1	$-0.280 \cdot 10^{-2}$	
$\mathbf{A}_{12}$	$-1 \cdot 10^{-3}$	-1	1	$-0.982 \cdot 10^{-3}$	
$\mathbf{A}_{21}$	$-1 \cdot 10^{-3}$	-1	1	$-1.074 \cdot 10^{-3}$	
$\mathbf{A}_{22}$	$-1 \cdot 10^{-2}$	-1	1	$-0.186 \cdot 10^{-2}$	
$\mathbf{B}_{11}$	$1 \cdot 10^{-3}$	-1	1	$1.709 \cdot 10^{-3}$	$J = 99.215$
$\mathbf{B}_{22}$	$1 \cdot 10^{-3}$	-1	1	$0.341 \cdot 10^{-3}$	$R^2 = 0.9971$
$\bar{\mathbf{x}}_1$	15	0	45	13.247	
$\bar{\mathbf{x}}_2$	1	0	45	5.353	
$\bar{\mathbf{u}}_1$	20	1	100	14.714	
$\bar{\mathbf{u}}_2$	25	1	100	28.416	
<i>SO model</i>					
$\mathbf{A}_{31}$	$-5 \cdot 10^{-5}$	-1	1	$-5.248 \cdot 10^{-5}$	
$\mathbf{A}_{32}$	$-2 \cdot 10^{-6}$	-1	1	$-18.47 \cdot 10^{-6}$	
$\mathbf{A}_{33}$	$-2 \cdot 10^{-2}$	-1	1	$-2.041 \cdot 10^{-2}$	
$\mathbf{A}_{34}$	$-2 \cdot 10^{-3}$	-1	1	$-2.034 \cdot 10^{-3}$	
$\mathbf{A}_{41}$	$-2 \cdot 10^{-6}$	-1	1	$-19.51 \cdot 10^{-6}$	
$\mathbf{A}_{42}$	$-1 \cdot 10^{-5}$	-1	1	$-3.369 \cdot 10^{-5}$	
$\mathbf{A}_{43}$	$-2 \cdot 10^{-3}$	-1	1	$-1.683 \cdot 10^{-3}$	$J = 110.579$
$\mathbf{A}_{44}$	$-2 \cdot 10^{-2}$	-1	1	$-1.958 \cdot 10^{-2}$	$R^2 = 0.9968$
$\mathbf{B}_{31}$	$-9 \cdot 10^{-6}$	-1	1	$31.60 \cdot 10^{-6}$	
$\mathbf{B}_{42}$	$-9 \cdot 10^{-6}$	-1	1	$6.125 \cdot 10^{-6}$	
$\bar{\mathbf{x}}_3$	15	0	45	13.314	
$\bar{\mathbf{x}}_4$	1	0	45	5.416	
$\bar{\mathbf{u}}_1$	20	1	100	14.857	
$\bar{\mathbf{u}}_2$	25	1	100	28.961	

Dataset has SNR=[8.6 3.8]

and thus the observed (measured) variables coincide with a subset of the state variables,  $T$  and  $C_A$ . In such situation: (i) the dimensions of matrices  $\mathbf{A}$ ,  $\mathbf{B}$  and  $\mathbf{C}$  (see (2.28)) are  $4 \times 4$ ,  $4 \times 2$  and  $2 \times 4$ , respectively; (ii)  $\mathbf{C}$  is constituted exclusively by 0 and 1 elements: the  $\mathbf{H}$  part of  $\mathbf{C}$  (see (2.31)) is the  $2 \times 2$  identity matrix; (iii) the two first rows of  $\mathbf{A}$  as well as the two first rows of  $\mathbf{B}$  are 0 except the elements  $\mathbf{A}_{13}$  and  $\mathbf{A}_{24}$  which are 1. For the

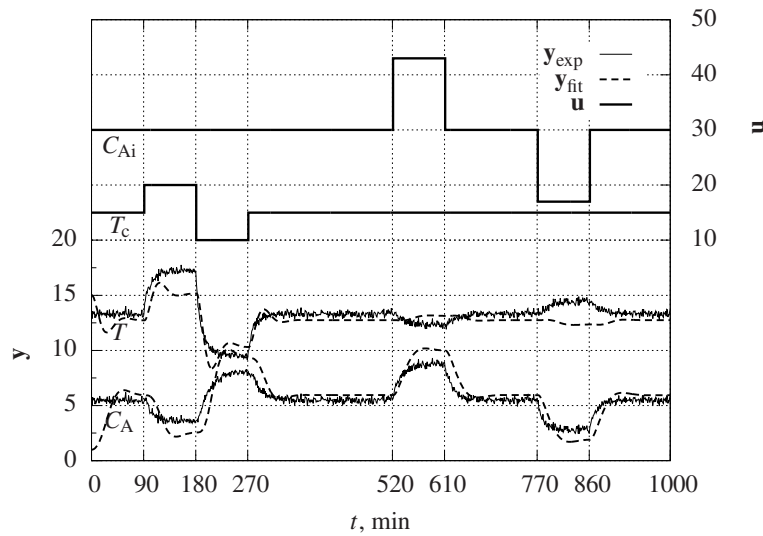


Figure 2.13: Difficulties in identifying the MIMO system via an SO model.

reasons also invoked when applying the FO model to this system, elements  $B_{31}$  and  $B_{42}$  were set to 0. The initial steady-state value for the state variables  $\hat{T}$  and  $\hat{C}_A$  was equally set to zero since both  $T$  and  $C_A$  are constant at steady-state. Therefore, the number of parameters needed to be estimated for the SO model applied to the MIMO system is 14.

According to (2.128), simultaneous accounting of both output curves of the MIMO system was considered during the optimization process (ie, the objective function was the sum of  $2 \times 1000$  square errors between original and predicted values), both when using the FO model (see above) or the SO model.

In the first attempt, the optimization algorithm encountered more difficulties in finding the parameters of this model. Even when the tolerance was decreased to  $10^{-10}$ , the resulting model presented bad prediction performance (Figure 2.13) with  $R^2 = 0.9323$  and  $J = 2430.979$ , which is frankly worse than that achieved with the FO model ( $R^2 = 0.9971$  and  $J = 99.215$ ). This unacceptable fit quality was caused by poor conditioning of the data. Since the tolerance values were already relatively close to the machine precision, the parameters were equally scaled up by a  $10^8$  factor, with the necessary changes in the model. This approach proved effective as the result-



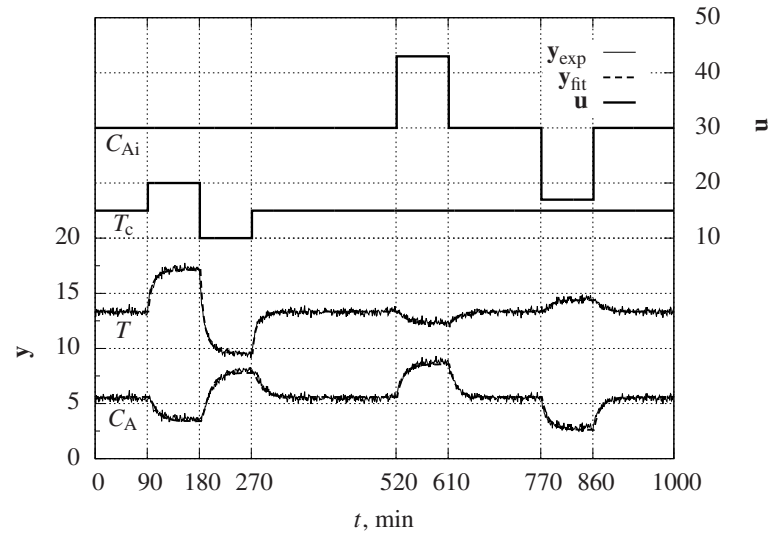


Figure 2.14: MIMO system identification using an SO model.

ing fit is as good as that obtained for the FO model. These parameters are listed in Table 2.2 and the corresponding model response can be observed in Figure 2.14. The SO model is now able to reproduce the system response in a comparable way to the FO model (compare Figures 2.14 and 2.12 and values of  $J$  and  $R^2$  in Table 2.2).

Since the performance of FO and SO models are comparable, the FO model is preferable as it represents the best trade-off between performance and simplicity.

## 2.4 Hybrid modeling of a biodiesel decanter<sup>4</sup>

One of the most relevant units of a biodiesel production line is the reactor, where the oil reacts with methanol under certain operating conditions to produce a mixture of biodiesel and the by-product glycerol. After the reaction, the mixture is cooled down and its components are separated. Figure 2.15 represents schematically the production line. It should be noted that the separation step in biodiesel industry is commonly performed in a gravity settler. The gravitational settling

<sup>4</sup>This section is based on Brásio, A. S., Romanenko, A., and Fernandes, N. C. (2015a). Development of a numerically efficient biodiesel decanter simulator. In *Operational Research, CIM Series in Mathematical Sciences 4*. Springer International Publishing Switzerland 2015. URL <http://www.springer.com/us/book/9783319203270>, with kind permission from Springer Science and Business Media B. V.

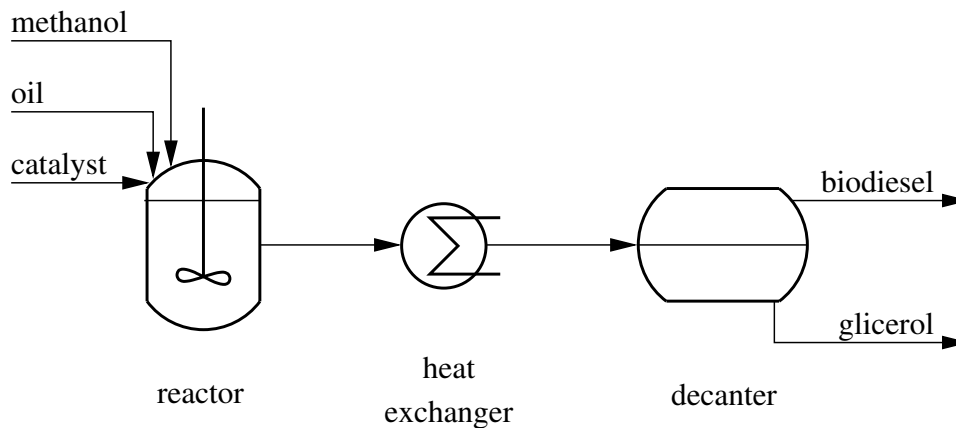


Figure 2.15: Simplified representation of the batch biodiesel production process.

is a lengthy process and therefore this step represents a significant part of the total production time, exceeding several times the residence time required in the reactor.

A decrease in the settling time would represent an economic process improvement. Thus, it is appealing to use dynamic optimization tools (Biegler, 2007) in order to reach a compromise between the objectives sought and the costs associated with them. These techniques are based on models that describe the dynamics of the process. Also, the operation of a biodiesel production line can be greatly improved by a system of non-linear predictive control based on first-principle models as described in Brásio et al. (2013) and Brásio et al. (2015).

In the decanter, two liquid phases coexist (the light and the heavy phases) that interact with each other. It is therefore necessary to model the liquid-liquid equilibrium in order to quantify this interaction in the dynamic model of the decanter. The quantification of liquid-liquid equilibria may be carried out by the *flash* calculation (Lobo and Ferreira, 2006), which is an iterative method.

However, a dynamic model which employs iterative methods cannot be integrated efficiently in a predictive control computing platform. In fact, the model is invoked dozens of times per iteration. Although the integrator has mechanisms to accelerate the convergence, the iterative calculation of phase equilibrium on each invocation of the model results in a significant computational burden and makes it more difficult to use of automatic differentiation tools, as ADOL-C (Walther and Griewank, 2012) or CppAD (Bell, 2012), because it significantly increases the memory needed to perform the calculations.

An alternative approach to the calculation of phase equilibrium in order to avoid the iterative method without deteriorating the quality of predictions is presented here. The results obtained by flash calculations are approximated by a model based on neural networks. Its type, composition and characteristics are detailed and its performance evaluated. By incorporating this data-driven model of the liquid-liquid equilibrium in the first-principle model of the decanter, a hybrid model is obtained. Such model describes accurately the underlying physical phenomena while it also ensures a feasible real time execution in the context of automatic differentiation.

### 2.4.1 Liquid-liquid equilibrium

The methodology most commonly used to quantify the liquid-liquid equilibrium between two partially miscible liquids is the *flash* calculation described in detail in Lobo and Ferreira (2006).

Considering a feed flow containing  $n_c$  components with composition  $x_{i,\text{in}}$ , the equilibrium at pressure  $P$  and temperature  $T$  is reached forming two distinct phases with composition  $x_{i,\text{lt}}$  and  $x_{i,\text{hv}}$ , respectively (where  $i = 1, \dots, n_c$ ). Thus, the feed is separated into two phases: the molar fraction  $L_{\text{lt}}$  constitutes the light phase and the remaining fraction  $1 - L_{\text{lt}}$  is the heavy phase. The equilibrium of each component in the mixture is set by  $K_i$  which represents the ratio of the molar fractions of chemical species  $i$  in the two liquid phases, i.e.

$$K_i = \frac{x_{i,\text{lt}}}{x_{i,\text{hv}}} = \frac{\gamma_{i,\text{lt}}}{\gamma_{i,\text{hv}}}, \quad (2.130)$$

where  $\gamma_{i,\text{lt}}$  and  $\gamma_{i,\text{hv}}$  are the activity coefficients of component  $i$  in the light and heavy phases, respectively.

Figure 2.16 represents schematically the mechanistic quantification of liquid-liquid equilibrium and shows its iterative nature. After specification of the feed, and already inside the iterative cycle, the UNIFAC method (or one of its variations) is used to determine the activity coefficients required to the calculation of the equilibrium constants. The UNIFAC method (Fredenslund et al., 1975) estimates the coefficients based on the sum of the contributions of functional groups present in the mixture components: ester, methanol and glycerol.

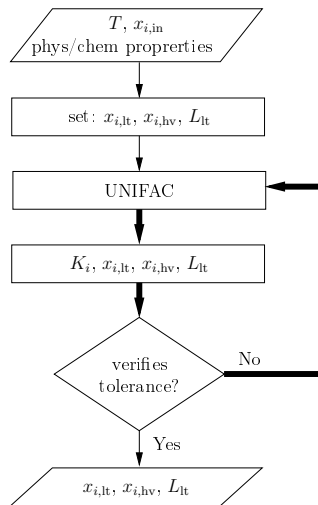


Figure 2.16: Flowchart of the *flash* method to determine the liquid-liquid equilibrium.

The oil that is the raw material for producing biodiesel is composed of glycerides (mainly triglycerides) whose skeleton consists of a glycerol molecule binding fatty acids. The oil has a biological origin and is characterized by natural variability. Typically, the lauric acid is the fatty acid in greater quantities in vegetable oils. For this reason and in the context of this study, it is considered that the fatty acid present in the raw material is lauric acid only (i.e., the ester contained in biodiesel is exclusively methyl laurate ester).

Once convergence for the *flash* calculation is reached, it is then possible to quantify the degree of separation of component  $i$  by the light and the heavy phases. From the amount initially present, the fraction of component  $i$  that goes to the light phase is given by

$$\xi_i = L_{lt} \frac{x_{i,lt}}{x_{i,in}} . \quad (2.131)$$

## 2.4.2 Dynamic mathematical model of a decanter

Consider now an industrial continuous decanter unit with parallelepipedic format and lying horizontally, as depicted in Figure 2.17.

The decanter inlet stream is the mixture that leaves the reactor flowing at a molar rate  $N_{in}$  and is characterized by composition  $x_{in}$  and temperature  $T$ .

In the decanter, all the components of the feed get split into two phases but in

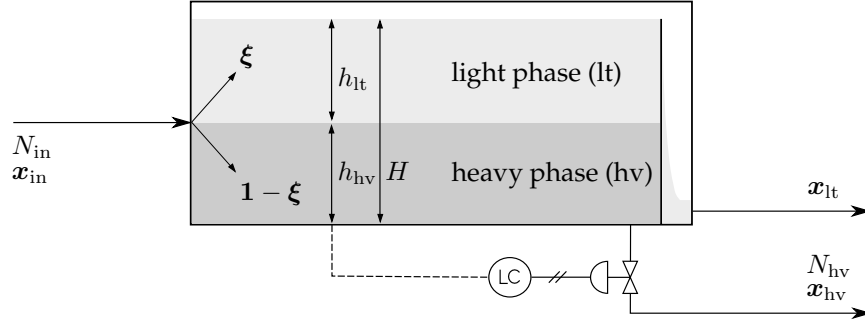


Figure 2.17: Schematic representation of the decanter.

different proportions from component to component. The degree of separation of a generic component  $i$  is quantified through the split fraction  $\xi_i$  which represents the fraction of component  $i$  that goes into the light phase. The set of the split fractions to the light phase for all the components is therefore the vector  $\xi = [\xi_E \ \xi_M \ \xi_G]$  and to the heavy phase is its complementary  $1 - \xi$ .

The decanter is equipped with an internal baffle. As the two phases separate, the heavy phase leaves the unit through its bottom while the light phase leaves the decanter by flowing over the baffle positioned close to its end. The dynamics of the subsection after the baffle may be neglected since its volume is insignificant compared to the total volume of the decanter. The output molar flow rate of the heavy phase,  $N_{hv}$ , is manipulated by a level controller.

The first principle mathematical model of this system includes partial and global mass balances and describes the evolution of the molar fractions of all the components in each of the phases as well as the heights of these phases. For a generic component  $i$  ( $i = E, M$ ),

$$n_{hv} \frac{dx_{i,hv}}{dt} = \sum_k^{n_c} ((1 - \xi_k) x_{k,in}) N_{in} \left( \frac{1 - \xi_i}{\sum_k^{n_c} ((1 - \xi_k) x_{k,in})} x_{i,in} - x_{i,hv} \right) \quad (2.132)$$

and

$$n_{lt} \frac{dx_{i,lt}}{dt} = \sum_k^{n_c} (\xi_k x_{k,in}) N_{in} \left( \frac{\xi_i}{\sum_k^{n_c} (\xi_k x_{k,in})} x_{i,in} - x_{i,lt} \right), \quad (2.133)$$

where  $n_{hv}$  and  $n_{lt}$  represent the amount of molecules in the heavy and light

phases, respectively. The composition of the remaining component (G) in phase  $j$  (with  $j = \text{hv}, \text{lt}$ ) is

$$x_{G,j} = 1 - \sum_i^{n_c-1} x_{i,j}. \quad (2.134)$$

The global molar balance to the heavy phase is

$$\frac{dn_{\text{hv}}}{dt} = \sum_i^{n_c} ((1 - \xi_i) x_{i,\text{in}}) N_{\text{in}} - N_{\text{hv}}. \quad (2.135)$$

The amount of molecules in the light phase is given by

$$n_{\text{lt}} = \frac{h_{\text{lt}} A}{\sum_i^{n_c} (V_i x_{i,\text{lt}})} \quad (2.136)$$

and the heights of both phases by

$$h_{\text{hv}} = \frac{n_{\text{hv}}}{A} \sum_i^{n_c} (x_{i,\text{hv}} V_i) \quad (2.137)$$

and

$$h_{\text{lt}} = H - h_{\text{hv}}, \quad (2.138)$$

where  $A$  is the area of the base of the decanter and  $V_i$  stands for the molar volume of component  $i$ .

The split fractions  $\xi$  are calculated using the previously developed neural network. Equations 2.136 and 2.137 assume that both phases are ideal. The physical properties that constitute the model parameters are specified in Table 2.3.

Table 2.3: Molar volume of ester, methanol and glycerol.

Component	$V$ ( $10^{-5} \text{ m}^3 / \text{mol}$ )
E	34.51
M	4.23
G	6.87

### 2.4.3 Neural network development and performance

Based on the composition and temperature of a mixture of ester, methanol and glycerol entering the decanter, the neural network must indicate how the three components are separated by the light and heavy phases, that is, must predict the split fractions to the light phase for all the components. Thus, the neural network has the temperature ( $T$ ) and the composition of the mixture as the input variables. The mixture composition is expressed in terms of molar fractions of methanol and of glycerol<sup>5</sup>,  $x_{M,in}$  and  $x_{G,in}$ . The output variables are the split fractions for the three components  $\xi_E$ ,  $\xi_M$  and  $\xi_G$ , indicating, for each component, the molar or mass fraction of the initial amount that goes to the light phase. The calculated split fractions are used to solve the mathematical model describing the decanter in a computationally efficient way.

#### ■ Generation and treatment of data

The training dataset was generated by the *flash* calculation described in Section 2.4.1.

The characterization of the feed mixture that enters the separation unit is especially important, since the network must be trained with a set of relevant data within the range usually observed in such systems. The authors of Bambase et al. (2007) experimentally performed the transesterification reaction of sunflower oil at 60°C using a molar ratio between methanol and oil of 6:1, 0.50 % ( $m/m$ ) of NaOH as catalyst and an agitation rate of 400 rpm. In that work, the component concentration over time is shown. However, the information about methanol, one of the components in largest quantity in the mixture, is omitted. For this reason, it was necessary to simulate the transesterification reaction (reactor) in order to obtain the dynamic profiles of the composition of all different chemical species required to fully quantify the mixture at the end of the reaction. The equilibrium and the speed of all the transesterification reactions are conditioned by reaction medium stirring. The work Brásio et al. (2011) proposes a methodology to explicitly include this variable in the model of the reactor. However, in the present context, a more simplistic model is enough to generate the datasets. The model and parameters described in the work Bambase et al. (2007) were used in

<sup>5</sup>Note that the molar fraction of the ester is linearly dependent of the molar fractions of the two other components.

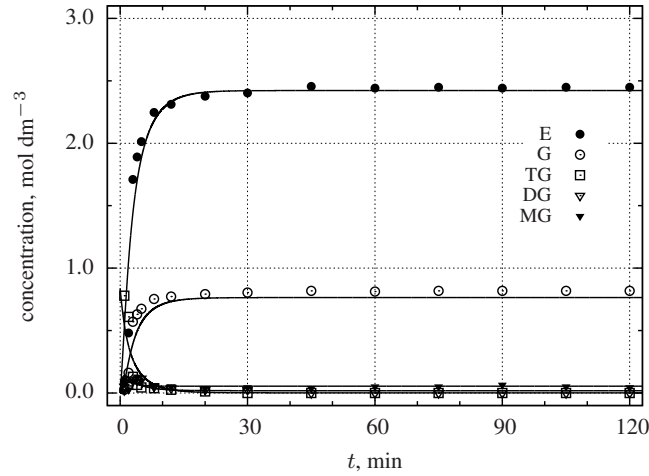


Figure 2.18: Simulation of the transesterification reaction of sunflower oil with model according to Bambase et al. (2007).

the simulation system and the corresponding results are shown in Figure 2.24. The visual comparison between simulated and experimental points as well as the obtained coefficient of determination ( $R^2 = 0.99998$ ) show a good match with the data of the system studied in Bambase et al. (2007).

The reaction mixture that leaves the reactor is directed to the decanter without undergoing any further change in its composition. Therefore, the simulation values of the reactor for the final time ( $t = 120$  min) correspond to the concentration values at the entrance of the decanter (mixture before separation). The molar concentration of the mixture to be separated is

$$\begin{aligned} \mathbf{C} &= [C_{TG} \ C_{DG} \ C_{MG} \ C_M \ C_G \ C_E] \\ &= [0.0018 \ 0.0188 \ 0.0550 \ 2.6181 \ 0.7644 \ 2.4219] \text{ mol/dm}^3. \end{aligned}$$

The fractions of tri-, di- and monoglycerides were considered to go to in the phase of the ester component (light phase) since their amounts are reduced and because the glycerides and the ester molecules have affinity. The corresponding molar fraction is given by

$$\begin{aligned} \mathbf{x} &= [x_E \ x_M \ x_G] \approx [x_E + x_{TG} + x_{DG} + x_{MG} \ x_M \ x_G] \\ &= [0.42 \ 0.45 \ 0.13]. \end{aligned}$$



Input Data		$\mu$	$\sigma$
$x_{M,in}$	– (n/n)	0.44291	0.05769
$x_{G,in}$	– (n/n)	0.14301	0.07483
$T$	°C	42.340	10.39

Table 2.4: Average ( $\mu$ ) and standard deviation ( $\sigma$ ) for the normalization of the input data.

The experimental dataset of the liquid-liquid equilibrium was generated at various temperatures of the inlet flow. For each temperature, a mesh was constructed by varying the molar fractions of the mixture. The following intervals were considered:  $25^\circ\text{C} < T < 60^\circ\text{C}$ ,  $0.32 < x_{E,in} < 0.52$  and  $0.35 < x_{M,in} < 0.55$ . The molar fraction of glycerol at the entrance of the decanter,  $x_{G,in}$ , was computed by the relation  $\sum_{i=1}^{n_c} x_{i,in} = 1$ . The range for the temperature was selected taking into account the typical reaction temperatures defined by Bambase et al. (2007). This range was then covered with increments of  $1^\circ\text{C}$ . The range of compositions was defined as  $\pm 0.10$  of the molar fractions  $x_{E,in}$  and  $x_{M,in}$  previously calculated. The defined range for compositions was split into intervals of 0.01. In total, 36 meshes of 405 points were generated.

The dataset normalization, of great importance in neural network (Chaturvedi, 2008), was performed using the values given in Table 2.4.

After the pre-treatment, the data were randomly divided into three sets: the training set, the validation set and the test set. The training and validation sets were used to estimate the parameters of the neural network. The test set was used to simulate the network allowing further comparison between the data obtained by the *flash* method and the prediction by the neural network.

### ■ Neural network specification

The neural network contains three distinct layers. The input layer has three neurons corresponding to the three input variables in the network  $\mathbf{X} = [x_{M,in} \ x_{G,in} \ T]^\top$ . An intermediate layer having five neurons and an output layer with three neurons corresponding to the variables  $\mathbf{Y} = [\xi_E \ \xi_M \ \xi_G]^\top$  are considered. Figure 2.19 graphically depicts the network structure.

The training algorithm from the software package `octave-nnet 0.1.13-2`

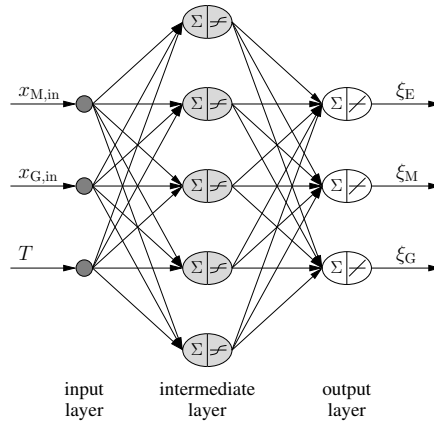


Figure 2.19: Neural network used to substitute the *flash* calculation.

for GNU Octave (Schmid, 2009) was used in the neural network training. The weights initialization is made using random elements uniformly distributed in the interval  $[-1, 1]$ . The initial learning rate  $\mu_0$  is set to  $10^{-3}$  (Hagan et al., 1996). Other parameters related to the training of the neural network were specified as follows: the maximum number of iterations was  $2 \times 10^3$ , the tolerance was  $5 \times 10^{-7}$  and the maximum time for training was  $10^3$  s .

### ■ Neural network training

The neural network with the described structure was trained. Figure 2.20 shows (see points) the evolution of the mean squared error over the training iterations. The Levenberg-Marquardt algorithm took 99 iterations to achieve the specified tolerance of  $5 \times 10^{-7}$  (indicated by the dashed line). The training process took 20 s. The network validation was done automatically by the software package. Figure 2.20 compares the MSE of the validation along iterations (solid line) with the MSE of the training dataset and shows a good fit between the two.

The resulting weighting matrices  $\mathbf{w}$  and  $\mathbf{W}$  are defined by

$$\mathbf{w} = \begin{bmatrix} -0.280399 & -1.089354 & -0.085569 & 0.139296 & 0.074241 \\ -0.133304 & -0.569687 & -0.994122 & 0.322463 & 1.155133 \\ -0.477709 & 0.281168 & 0.030668 & -0.061436 & -0.017357 \end{bmatrix}$$

and

$$\mathbf{W} = \begin{bmatrix} 1.6266 \times 10^{-4} & 1.7624 \times 10^{-4} & 8.4364 \times 10^{-3} & 1.2702 \times 10^{-4} & 4.2978 \times 10^{-3} \\ 5.7984 \times 10^{-3} & 7.9335 \times 10^{-3} & 2.2665 \times 10^{-0} & -2.9864 \times 10^{-1} & 7.3118 \times 10^{-1} \\ -1.3643 \times 10^{-3} & 7.3683 \times 10^{-4} & 3.9128 \times 10^{-1} & 2.6466 \times 10^{-3} & 1.5635 \times 10^{-1} \end{bmatrix}$$

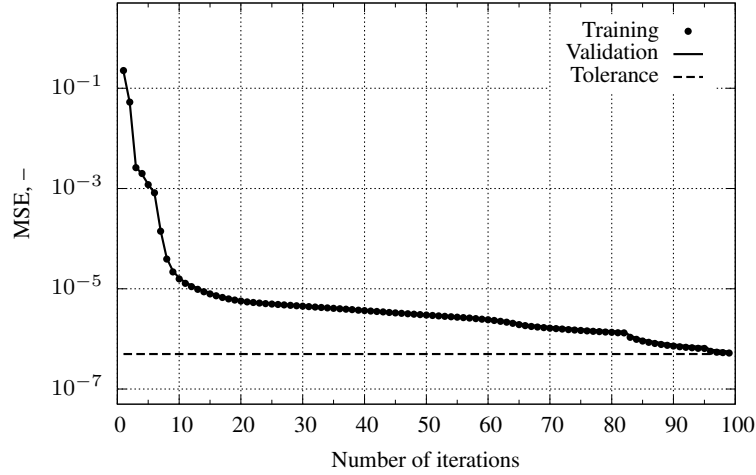


Figure 2.20: Evolution of the average square error while applying the Levenberg-Marquardt algorithm.

and the bias vectors  $\theta$  and  $\Gamma$  correspond to

$$\theta = \begin{bmatrix} 0.62280 \\ 1.15884 \\ -2.46359 \\ 0.27300 \\ 2.24627 \end{bmatrix} \text{ and } \Gamma = \begin{bmatrix} 1.00373 \\ 1.89116 \\ 0.23502 \end{bmatrix}.$$

The comparison of the computational efforts reveals that the average time required to generate a point by the *flash* calculation was about 0.018 129 s, while using the neural network was approximately 0.000 129 s. Therefore, a speedup of 141 times was achieved by the neural network over the *flash* calculation.

### ■ Predicting capability of the network

Figure 2.21 shows the prediction of the split fractions using the neural network. It also includes the first 250 points of the test. Methanol is the component with the biggest variation of its split fraction within the range of temperature and composition covered by the mesh. Regarding the ester, it is the component for which the split fraction is less dependent on the initial conditions of the mixture (ie, its composition and temperature). In fact, as it can be seen in Figure 2.21, more than 99.9% of the ester always goes to the light phase, regardless of the initial condi-

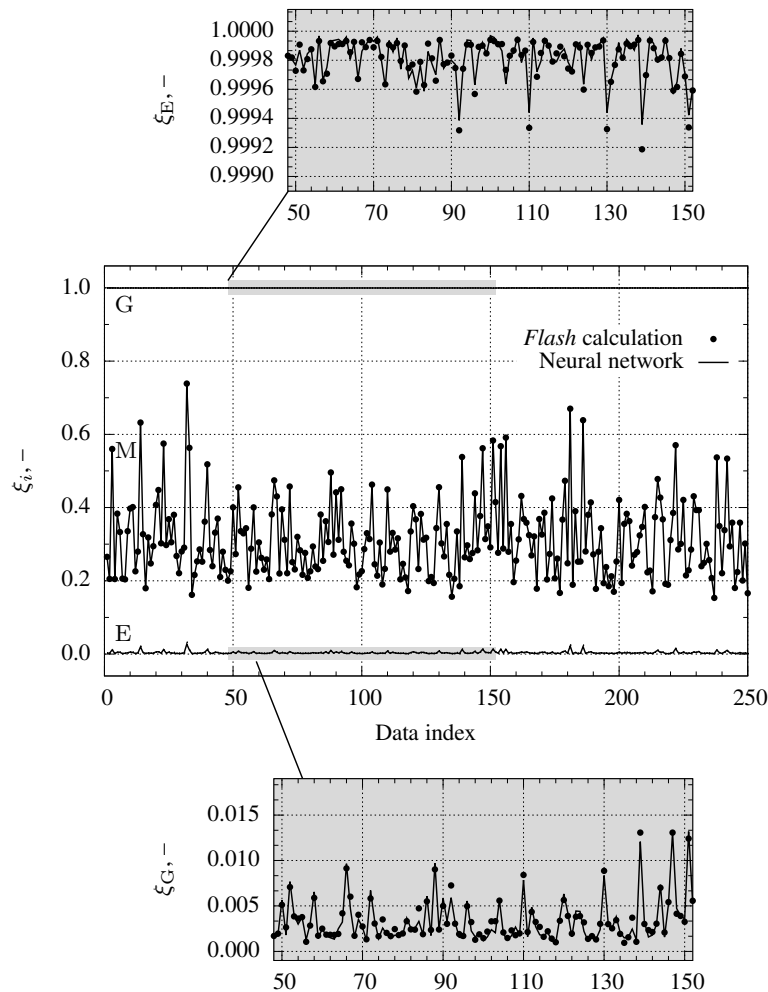


Figure 2.21: Prediction of the split fractions through the neural network model.

tions of the mixture to be separated. Finally, the variation of the split fraction of glycerol as a function of composition and temperature of the decanter feed mixture is also light and it migrates almost entirely to the heavy phase. In order to allow for a better understanding of the data (Figure 2.21), two areas of the main graph were zoomed in, one relative to the data for the glycerol component and other to data concerning the ester component.

The determination coefficient values corresponding to the estimates of methanol, ester and glycerol are  $R^2(\xi_M) = 0.9999$ ,  $R^2(\xi_E) = 0.9137$  and  $R^2(\xi_G) = 0.9676$ , respectively. The prediction is especially good in the case of methanol, since this component is more sensitive to the initial conditions of the mixture. However,

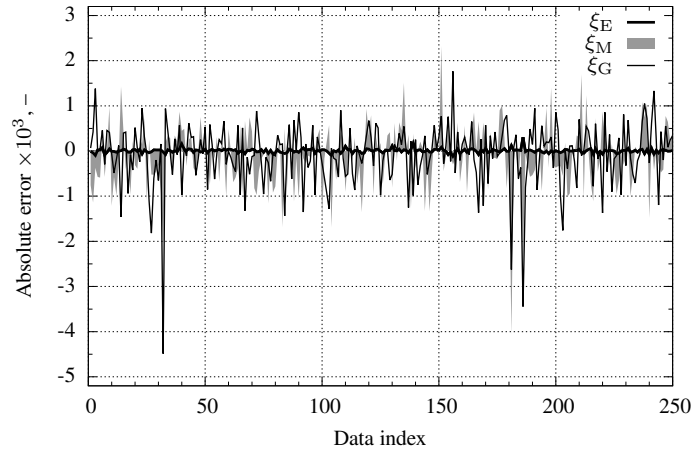


Figure 2.22: Absolute error between the predictions of the neural network and the of the *flash* calculation.

although the determination coefficients for glycerol and for ester are somewhat lower, the absolute errors between the predictions and the experimental values are quite low (see Figure 2.22).

The effect of temperature on the liquid-liquid equilibrium is quite pronounced. In order to show that the neural network is capable of predicting this effect, two equilibria corresponding to two mixtures A and B with different compositions under different temperatures were studied.

Mixture A is characterized by a molar fraction  $\mathbf{x}_{in} = [0.42 \ 0.45 \ 0.13]$  in accordance with the experimental values of Bambase et al. (2007). The second study deals with mixture B resulting from a greater reaction yield than the one verified for mixture A. Based on this assumption, mixture B was defined as having the composition  $\mathbf{x}_{in} = [0.47 \ 0.43 \ 0.10]$ .

Figure 2.23 represents the split fractions predicted by the *flash* calculation and by the neural network as functions of the temperature, showing a good match.

As discussed above, the split fraction of methanol varies significantly with temperature, in opposition to the fractions of ester and glycerol that remain approximately constant. A considerable zoom in of the graphical representation of these two fractions (see Figure 2.23) reveals what, at a first glance, could be considered as a discrepancy, particularly in the case of the ester. However, this difference is less than 0.006% (6 thousandths percent), and therefore negligible.

An increase in the temperature, the methanol and the glycerol split fractions

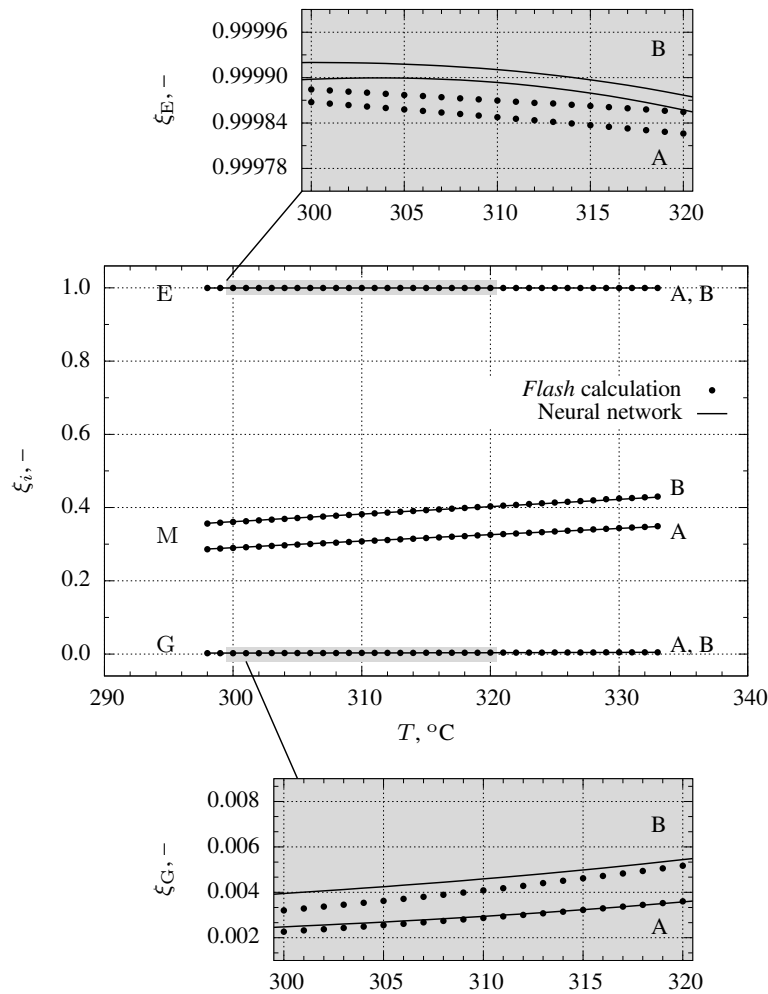


Figure 2.23: Prediction of the split fractions as function of temperature for two different mixtures A and B (A:  $[0.42 \ 0.45 \ 0.13]$ , B:  $\mathbf{x}_{in} = [0.47 \ 0.43 \ 0.10]$ ).

increase in both mixtures, although with less intensity in the case of glycerol. In mixture B (mixture richer in ester), methanol is more soluble in the light phase and, therefore, the split fraction is greater than the one obtained in mixture A (compare  $\xi_M$  for mixture A and B in the main plot of Figure 2.23). A similar effect is observed for glycerol (see zoom in of  $\xi_G$  in Figure 2.23). Conversely, the ester becomes more soluble in the heavy phase and, therefore, its split fraction to the light phase decreases (see zoom in of  $\xi_E$  in Figure 2.23).

### 2.4.4 Discussion of results

The developed neural network was used to quantify the interaction between the two liquid phases by calculating the split fractions for all the components in the context of the dynamic modeling of a decanter in a CPU time efficient way.

Suppose that, at initial time, the continuous decanter with dimensions  $1\text{ m} \times 1\text{ m} \times 3\text{ m}$  is filled with equal volumes of glycerol and ester. This combination forms two immiscible liquid phases with glycerol at the lower layer due to its higher density. Therefore, the initial height of the heavy phase is  $h_{\text{hv}} = 0.5\text{ m}$  and the initial height of the light phase is  $h_{\text{lt}} = 0.5\text{ m}$ . In such conditions, the initial phase compositions are  $\mathbf{x}_{\text{lt}} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$  and  $\mathbf{x}_{\text{hv}} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$ . At the same initial instant, the reaction mixture is fed to the decanter with a flow rate of  $N_{\text{in}} = 9.67\text{ mol s}^{-1}$ , composition  $\mathbf{x}_{\text{in}} = \begin{bmatrix} 0.42 & 0.45 & 0.13 \end{bmatrix}$  (corresponding to the aforementioned mixture A), and temperature  $T = 60^\circ\text{C}$ .

The heavy phase level  $h_{\text{hv}}$  is controlled through a PI controller using the molar flow rate  $N_{\text{hv}}$  as manipulated variable (initialized at  $0\text{ mol s}^{-1}$ ). The controller was tuned by the trial-and-error method with  $K_C = -500\text{ mol s}^{-1}\text{ m}^{-1}$ ,  $\tau_I = 2000\text{ s}$ , and  $\tau_D = 0\text{ s}$ .

#### ■ Operation start-up

The decanter start-up operation is simulated along a time horizon of 20 h with a time interval of 10 s. Figure 2.24 exhibits the dynamic response of the unit. As soon as the feed is introduced, the compositions of the light and of the heavy phases change due to the entrance of new components.

The split fractions computed by the neural network allow to define the affinity that each component will have to each of the heavy and light phases. For the feed conditions listed above, the split fractions are  $\boldsymbol{\xi} = \begin{bmatrix} 0.9998 & 0.3481 & 0.0046 \end{bmatrix}$ . Remark the high split fraction to the light phase for ester and the low split fraction for glycerol.

Methanol is attracted by both phases originating changes in their composition in what concerns this component. Glycerol does not have much affinity to the light phase and, as result, its molar fraction remains near zero in this phase. Conversely, the ester goes almost exclusively to the light phase and, therefore, the

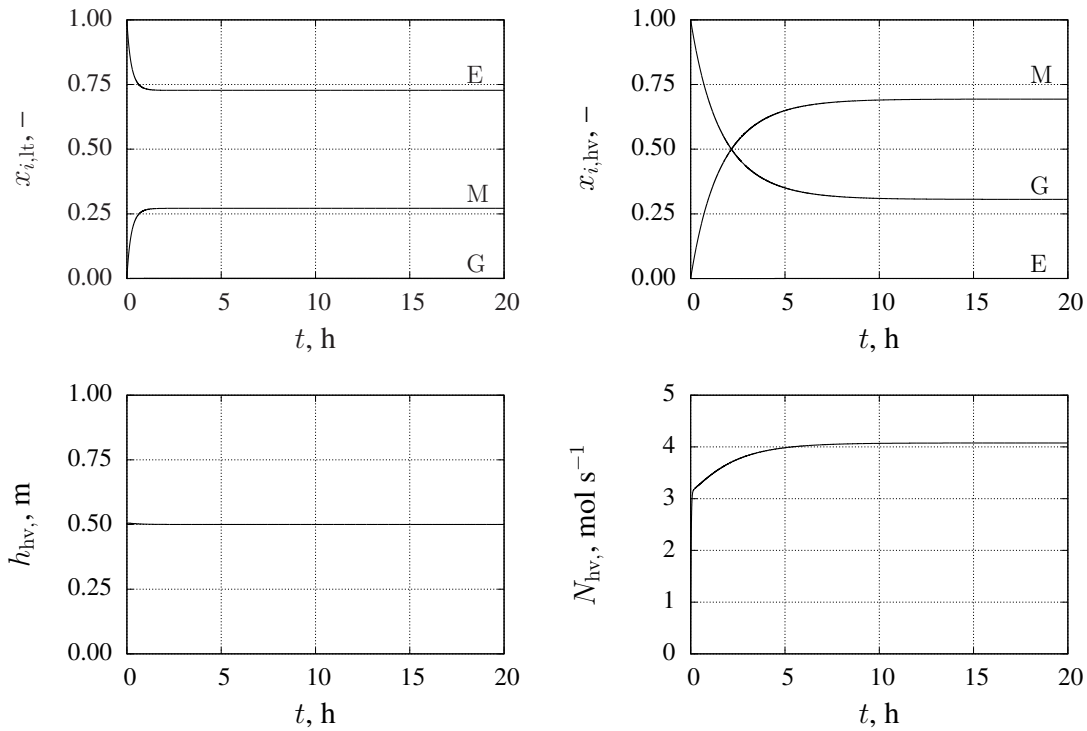


Figure 2.24: Profiles of the state variables and molar flow rate of the heavy phase under the start-up of the decanter operation.

composition of ester in the heavy phase remains approximately zero.

After approximately 10 h, the decanter reaches a steady-state with a composition of the light phase  $\mathbf{x}_{lt} = [0.728 \quad 0.271 \quad 0.001]$  and a composition of the heavy phase  $\mathbf{x}_{hv} = [0.000 \quad 0.694 \quad 0.306]$ .

From the graphs of Figure 2.24 it is also evident that the light phase has a much faster dynamics than the heavy phase. In spite of the fact that the volumes of both phases are the same throughout the experiment ( $0.5 \text{ m}^3$ ), the light phase is crossed by a volumetric flow 7 times bigger than the volumetric flow crossing the heavy phase. As a consequence, the residence time in the light phase is much smaller resulting in a faster dynamic response.

As it is clear from Figure 2.24, the level is kept by the controller at the setpoint of 0.5 m during the whole test increasing the output molar flow rate  $N_{hv}$  from zero until it finally stabilizes at  $4.08 \text{ mol s}^{-1}$ .



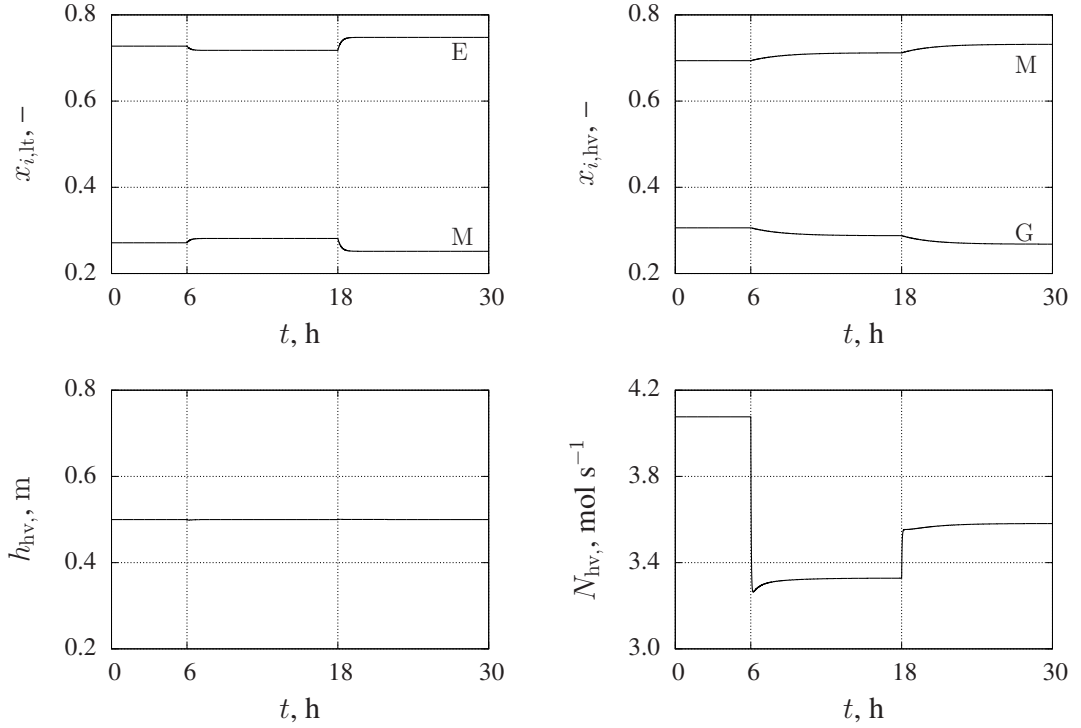


Figure 2.25: Profiles of the state variables and molar flow rate of the heavy phase under disturbances to the operation.

### ■ Introduction of disturbances

The process dynamics of the decanter affected by disturbances is illustrated below. The system, reinitialized at the steady-state encountered during the study of the system start-up, is subjected to various disturbances at instants  $t = 6$  h and  $t = 18$  h. Figure 2.25 depicts the evolution of key variable describing the system behavior in such situations. The ester composition in the heavy phase and the glycerol composition in the light phase were omitted from the graphs because they remain very low (approximately zero) along the whole test.

At instant  $t = 6$  h, the mixture that constitutes the feed is replaced by a mixture richer in ester (that is, the feed is changed from mixture A to mixture B). Therefore, the feed composition changes to  $\mathbf{x}_{in} = [0.47 \ 0.43 \ 0.10]$ . In view of this new condition, the neural network foresees a new liquid-liquid equilibrium and, in accordance, updates the split fractions to  $\boldsymbol{\xi} = [0.9998 \ 0.4283 \ 0.0069]$ . It is worth mentioning that the ester and glycerol split fractions for the light phase do not suffer significant changes. However, the methanol split fraction increases

substantially.

As Mixture B is poorer in methanol than mixture A, the amount of methanol going to both phases inside the decanter is pushed down. However, the new feed originates, in parallel, a bigger split fraction to the light phase for methanol. This induces a bigger amount of methanol going to the light phase. This second effect overlaps the first and, in consequence, the amount of methanol going to the light phase increases as a result of the disturbance introduced at  $t = 6$  h. The total molar amount moving into the light phase also increases as a consequence of this disturbance (because of methanol but, especially, because of ester). Although this fact tends to reduce the molar fraction, the increase in the amount of methanol is enough to impose an increase in methanol molar fraction, as shown by Figure 2.25. The amount of methanol going to the heavy phase decreases as a result of the introduced disturbance. However, since the total molar amount going to the heavy phase decreases (because of smaller methanol and glycerol contributions), the molar fraction of methanol increases as Figure 2.25 reveals.

The amount of ester flowing to the light phase increases, but its molar fraction decreases due to the more significant effect of the overall amount increase in the light phase (namely methanol and ester). In what concerns the molar fraction of glycerol in the heavy phase, it diminishes (see Figure 2.25). On one hand the amount of this component passing to the heavy phase is less and, on the other, the total molar amount of the heavy phase is higher.

To keep the level at the setpoint, the flow rate  $N_{hv}$  is changed. Once the rates of methanol and glycerol sent to the heavy phase are smaller, the controller has to lower the flow  $N_{hv}$  in order to be able to keep the level at its setpoint.

After a steady-state is reached, at  $t = 18$  h the feed temperature is reduced from  $60^\circ\text{C}$  to  $30^\circ\text{C}$ . This disturbance changes again the component distribution (the split fraction becomes  $\xi = \begin{bmatrix} 0.9999 & 0.3677 & 0.0041 \end{bmatrix}$ ). For these new operating conditions, the fraction of the inlet methanol that goes to the heavy phase is higher, inducing an increase of the methanol molar fraction and a larger glycerol dilution, that is, a decrease in glycerol molar fraction, in the heavy phase.

At the same time, the methanol molar fraction to the light phase decreases. Consequently, a smaller rate of methanol is directed to this phase whilst the molar rates of the other two components remain practically unchanged. Therefore, the

molar fraction of methanol and ester in the light phase augments and diminishes, respectively.

The level controller increases again the flow rate since the rate of methanol sent to the heavy phase also increases as a result of this second disturbance.



# Chapter 3

## Control Loop Performance Assessment

This chapter addresses the issue of control loop performance assessment, starting with a comprehensive and critical review on the subject. Because stiction is a long-standing control valve problem, a systematic taxonomy of the existing contributions is provided and techniques for its correction detection are developed: a method for the detection and characterization of oscillations (stiction consequences), a method for stiction detection and quantification based on numerical optimization and able to handle discontinuities of the model describing stiction, and a method for stiction detection in integrating processes based on pattern recognition.

### 3.1 Importance and characteristics

A control system is a set of interconnected components under certain configuration as represented in Figure 3.1. The controller task is to maintain the measured process variable at a specified setpoint value, in spite of disturbances acting on the process, problems in the actuators, noise affecting the sensors and/or changes in setpoint values. A satisfactory process control may only be achieved when the components of the control loop are working properly. It is obvious, even for single loop control systems, that the task of maintaining all components healthy is not trivial in a plant that comprises hundreds to thousands control loops.

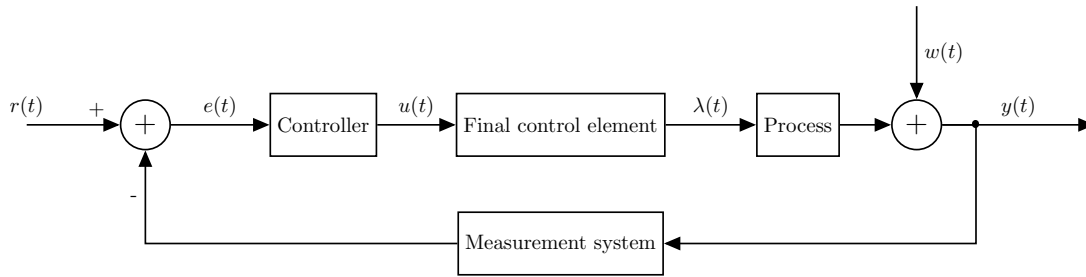


Figure 3.1: Block diagram of a single loop system, where  $r(t)$  is the set-point/reference variable (SP),  $e(t)$  is the control error,  $u(t)$  is the controller output or manipulated variable (MV),  $\lambda(t)$  is the real position of the final control element (OP),  $w(t)$  is the process disturbance, and  $y(t)$  is the controlled variable or process variable (PV).

The primary objective of control systems is to conduct the production processes in order to transform raw materials into products maximizing the profits while satisfying a series of performance criteria, such as product quality specifications, operational constraints, safety, and environmental regulations. In the commissioning phase, the design and implementation of control strategies and their tuning are carried out achieving a good level of performance. However, after some time in operation, the production process is exposed to changes in the raw material characteristics, modifications of the operation conditions and changes in the state of equipments (such as aging, wear, fouling and physical modifications) which may lead to performance degradation. Even well-designed control loops may experience problems due to sensors and actuators difficulties calling for the retuning of the controllers (Seborg et al., 2010; Jelali, 2010).

In order to detect performance deterioration, control loops should be supervised. Traditionally, this task was carried out manually by the plant personnel. However, during the last decades, a drastic personnel downsizing has occurred along with increasing demands on product quality, productivity and environmental impact that force companies to operate at a top performance. Therefore, control systems able to deliver high performance have been increasingly recognized as capital assets that should be maintained, supervised and revised routinely and automatically with the aid of control performance monitoring and assessment technologies (Jelali, 2010).

Although other types of supervisory control are increasingly used (such as

advanced control approaches), they are organized hierarchically providing the reference variables to low level regulatory controllers, typically PID controllers. Thus, the performance of the overall process relies on the PID controllers performance. For this reason, the development of tools that supervise/assess automatically their performance and detect common issues affecting performance before they become significant problems — the so-called loop performance monitoring and assessment tools — is quite important.

While assessment refers to the evaluation of some metrics, monitoring consists in following closely those metrics to detect eventual changes. In spite of these more rigorous definitions, the two terms are used interchangeably in the common jargon of industry and even in the technical literature. Independently of the names adopted, these tools must provide an online automated procedure that gathers large quantities of information to determine and evaluate the performance of the control system, freeing control engineers for higher valued tasks. This technology is characterized by the use of raw data, gathered in a non-invasive way and implemented in a completely automatic mode, and allows the detection and diagnosis of problematic or under-performing control loops (Salahshoor et al., 2011). Another important issue for monitoring/assessment tools is the appropriateness of the human-machine interface which should warranty a convenient presentation of results to the user. Still the same authors also emphasize the possibility of faults (such as false and missed alarms) that tend to reduce the user's trust on the tool. These tools are supposed to provide information succinctly through single page summaries and time trends to assist the problem diagnosis (Desborough and Miller, 2002). Other typical elements of these tools are a prioritized list of control loops with poor performance, with a special focus on PID controllers, as well as on-demand analysis.

According to Jelali (2006), the monitoring and assessment tool should include five main stages (see Figure 3.2). Once determined the capability of the running control system, the metrics/benchmarks for the performance monitoring and assessment are selected/designed. At this point, the detection of the poor performing control loops and the diagnosis of their underlying causes is carried out. The assessment process closes suggesting actions in order to improve the performance of the system.

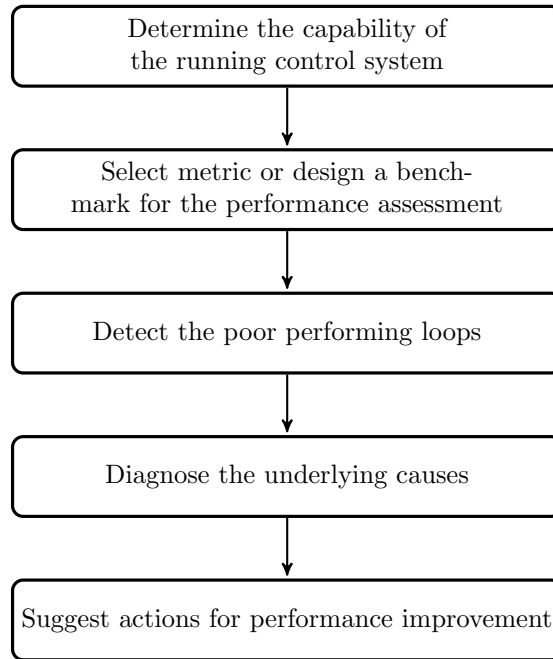


Figure 3.2: Control performance assessment stages.

## 3.2 State of the art

This section documents the current state of the art related to industrial controller performance assessment from the poor performance identification/detection to its diagnosis.

### 3.2.1 Poor performance detection

Performance assessment of regulatory control loops is widely documented because of their importance in a plant control hierarchy. In general terms, the performance of a regulatory control loop is related with its ability to deal with deviations between controlled variables and their setpoint values. The performance index allows to express the degree of performance with a single value and there is a range of techniques, based on different criteria, for its calculation. For example, Qin (1998) classification distinguishes deterministic performance criteria (traditional performance measures used in the case of deterministic disturbances) from stochastic performance criteria (criteria directly related to product quality and energy or material consumption). Shardt et al. (2012) system-



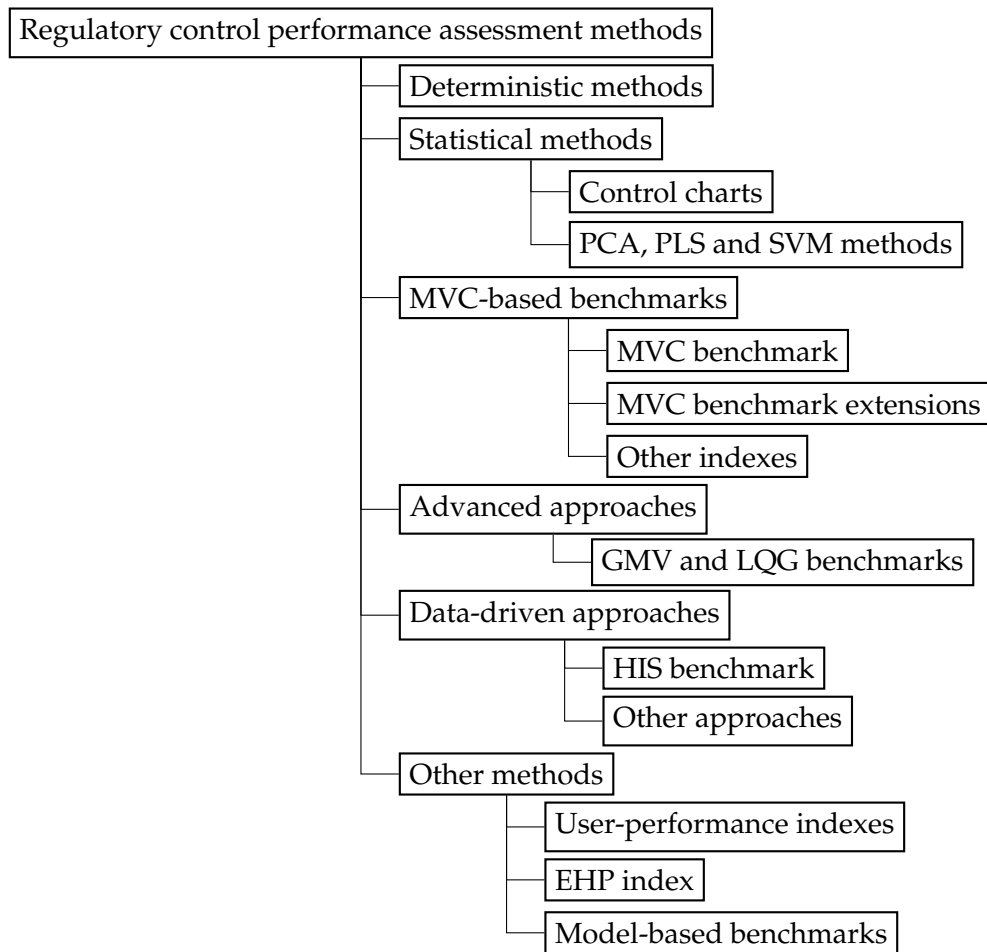


Figure 3.3: Regulatory control performance assessment methods.

atized the different techniques into statistical methods, MVC benchmark, LQG benchmark, and data-driven approach. The remainder methods, less significant, were lumped together under the designations of “other performance methods”. The review here performed merges these two classifications and slightly recasts them to incorporate techniques not mentioned there. The proposed taxonomy tree is shown in Figure 3.3. Table 3.1 condenses the information related to the suggested groups, namely in what concerns most representative works and advantages/disadvantages of that approaches. A more detailed description of each of these groups follows.

Topic(s)	Advantages	Disadvantages
<b>Deterministic methods</b> (Farenzena, 2008; Goodwin et al., 2001)	Clear conclusions about the performance and the robustness	Expensive to calculate in real time
<b>Statistical methods</b> Control charts (Cinar and Undey, 1999; Bersimis et al., 2007) PCA (Smith, 2002; Bersimis et al., 2007; Shlens, 2009)  PLS (Bersimis et al., 2007; Zhang and Zhang, 2010)  SVM (Sun and Tsung, 2003; Joachims, 2005; Lu et al., 2010)	Graphical tools, easy interpretation  Easily applicable to multivariate systems  Easily applicable to multivariate systems  Applicable to nonlinear and multivariate systems	Not easily applicable to multivariate systems  Missing data and noise are obstacles, problems in scaling, uses linear relationships  Missing data and noise are obstacles, problems in scaling, uses linear relationships  Problems in scaling, may become computationally intensive
<b>MVC-based benchmarks</b> (Harris, 1989; Huang et al., 2005)	Easy to calculate, minimal information required	May generate different interpretations
<b>Advanced benchmarks</b> (Grimble, 2002; Huang, 2003; Kozub, 2002; Danesh Pour et al., 2009)	Gives an absolute benchmark for performance improvement	May become computationally intensive
<b>Data-driven approach</b> (Huang et al., 2005, 2006; Huang and Kadali, 2008)	No model is required, easy to calculate	Selecting data for comparison may be problematic
<b>Other methods</b> (Hugo, 2006; Ingimundarson, 2003, 2006)	No model is required, easy to calculate	Selecting data for comparison may be problematic

Table 3.1: Overview of the regulatory control performance assessment methods.

### ■ Deterministic methods

Deterministic metrics (also called classical performance metrics) provide clear conclusions about the performance and the robustness of a control loop. However and in contrast to the others, these metrics are very expensive to calculate in real time because they usually need information from intrusive tests.

For stable systems, the control loop performance can be measured by param-

eters that describe the system dynamics such as the rise time, the settling time, and the overshoot (Farenzena, 2008; Goodwin et al., 2001). Figure 3.4, which exhibits the unitary step response of a system in closed-loop, helps to define these concepts. The rise time is the elapsed time up to the instant at which 95% of the steady-state value is reached for the first time. The settling time is the elapsed time until the step response gets confined inside a specific deviation band around the steady-state value. In Figure 3.4, the band represented in grey filled area corresponds to  $y_{\infty} \pm 5\%$ . Finally, the overshoot is the maximum amount by which the step response exceeds its final value (usually expressed as a percentage of the steady-state value). Based on these metrics, ratios between rise times and settling times obtained for open- and closed-loop responses may also be computed.

As for the measurement of the control loop robustness, metrics as the gain margin, the phase margin and the maximal sensitivity may be used (Farenzena, 2008). These metrics provide information on how far from the stability the current control loop is. The gain margin is defined as the maximal additional gain that the closed-loop would take to reach the critical condition. Similarly, the phase margin represents the pure phase delay that could be added to achieve the critical point. And the maximal sensitivity measures the largest amplification of the closed-loop response to process uncertainties. A figure plotting the amplitude ratio (in a logarithmic scale) and the phase of the frequency response of the system on a logarithmic frequency scale (called Bode diagram) easily shows both phase and gain margins (Figure 3.5a). A plot representing the imaginary versus the real part of the system transfer function (called Nyquist plot) gives an indication of the maximal sensitivity, as portrayed in Figure 3.5b. According to the Bode stability criterion, the system is stable because the amplitude ratio at the critical frequency is less than one.

### ■ Statistical methods

Statistical process control is related to the monitoring of process statistics using control charts to determine if the process is behaving properly, that is, if it is “in control”. Among the most commonly used are the mean, the standard deviation (or variance) and the range of the control error  $e(t)$  and of the manipulated variable  $u(t)$ . In general, a control chart is composed by a center line (CL)

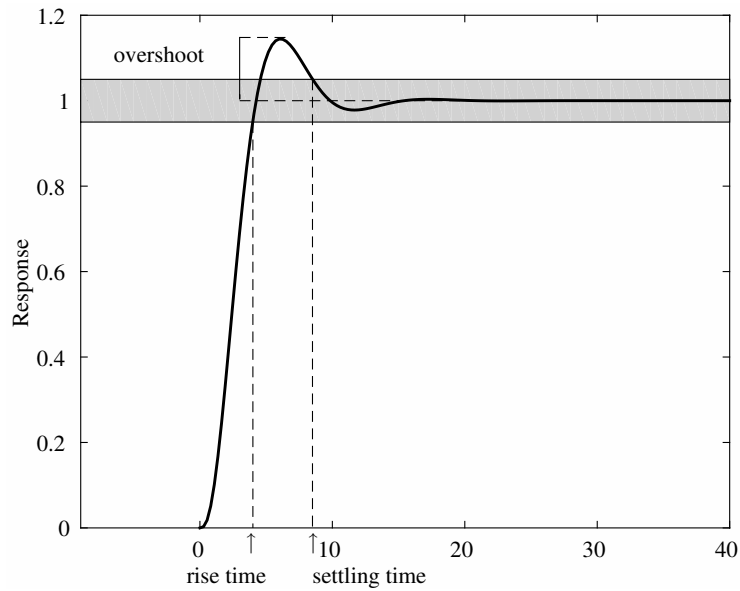


Figure 3.4: Deterministic metrics for performance evaluation in the feedback loop defined by  $H(s) = G(s)/(1 + G(s))$  with  $G(s) = 1/(s^3 + 3s + 2s)$ .

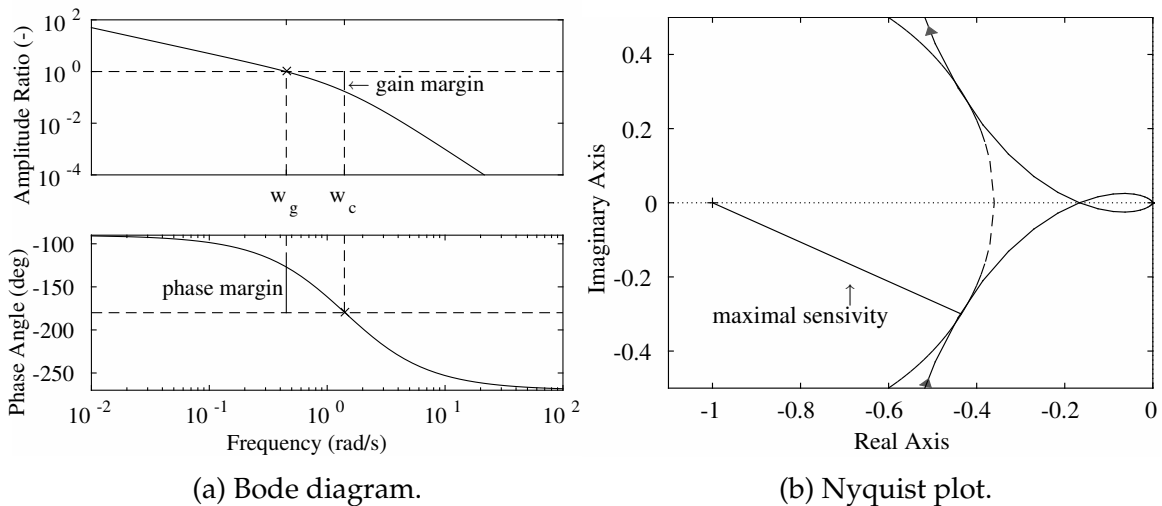


Figure 3.5: Robustness metrics for performance evaluation in the feedback loop defined by  $H(s) = G(s)/(1 + G(s))$  with  $G(s) = 1/(s^3 + 3s + 2s)$ .  $w_c$  stands for critical frequency and  $w_g$  for gain-crossover frequency.

representing the mean value for the in-control process and two other horizontal lines showing the upper control limit (UCL) and the lower control limit (LCL). Whether the represented variable in the control chart is normally distributed or not, it is usually acceptable to set the control limits to a multiple of the standard deviation. Figure 3.6 depicts the basic elements of a control chart with a generic sample statistic  $w$ .

There are different control charts, classified into univariate and multivariate depending on the number of process characteristics under supervision, as displayed in Figure 3.7. Shewhart control charts, cumulative sum control charts, and exponentially weighted moving average control charts are specially indicated for univariate process control.

**Shewhart control charts** They are charts based on the mean, the standard deviation, or the range.

The first ones, called Shewhart  $\bar{X}$  control charts, consider the characteristics

$$\text{UCL} = \bar{w} + k\sigma_w \quad (3.1a)$$

$$\text{CL} = \bar{w} \quad (3.1b)$$

$$\text{LCL} = \bar{w} - k\sigma_w \quad (3.1c)$$

where  $\bar{w}$  is the in control process variable mean and  $k$  is the distance between the control limits and the center line expressed in terms of standard deviation  $\sigma_w$  (usually set to the value 3). The standard deviation, usually unknown, can be estimated in two alternative ways:

- based on mean sample standard deviation  $\bar{s}_w$  as

$$\sigma_w = \frac{\bar{s}_w}{c_4}, \quad (3.2)$$

$$c_4 = \sqrt{\frac{2}{n-1} \frac{(\frac{n}{2}-1)!}{(\frac{n-1}{2}-1)!}}, \quad (3.3)$$

where  $c_4$  is a group size dependent function and the mean sample standard deviation for each of the  $m$  preliminary groups of size  $n$  is given

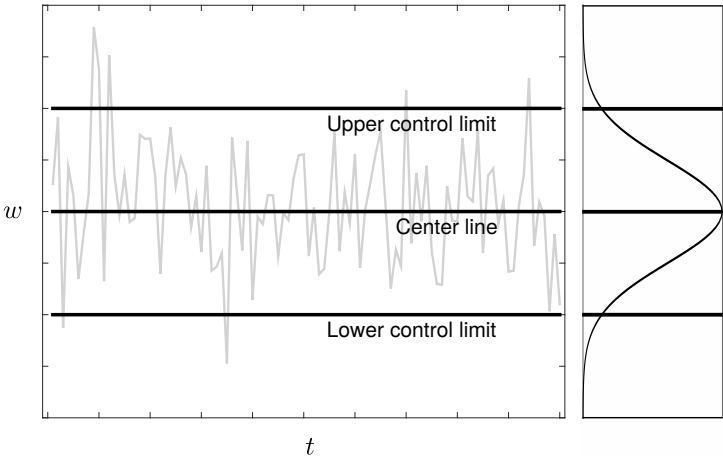


Figure 3.6: Control chart representation.

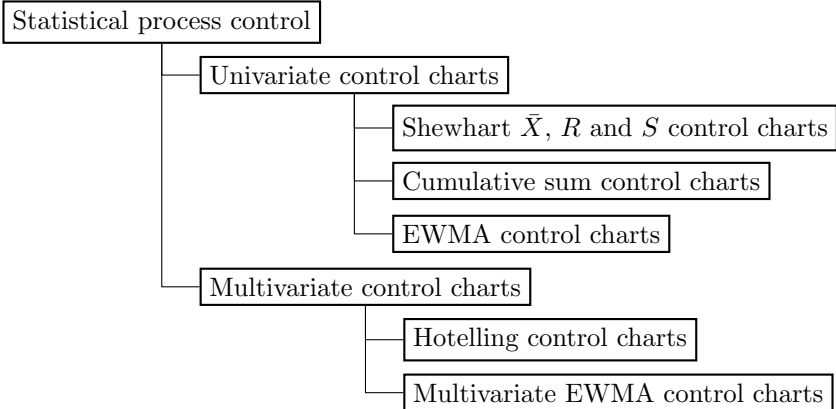


Figure 3.7: Control charts used in statistical process control.

by

$$\bar{s}_w = \frac{1}{m} \sum_{i=1}^m s_{wi}, \quad (3.4)$$

$$s_{wi} = \sqrt{\frac{\sum_{j=1}^n (w_{ij} - \bar{w}_i)^2}{n-1}}. \quad (3.5)$$

where  $s_{wi}$  represents the sample standard deviation for group  $i$ . With this information, it is possible to determine the Shewhart  $\bar{X}$  control chart properties (3.1) by

$$\text{UCL} = \bar{w} + k \frac{\bar{s}_w}{c_4 \sqrt{n}} \quad (3.6a)$$

$$\text{CL} = \bar{w} \quad (3.6b)$$

$$\text{LCL} = \bar{w} - k \frac{\bar{s}_w}{c_4 \sqrt{n}} \quad (3.6c)$$

- based on mean range  $\bar{R}_w$  as

$$\sigma_w = \frac{\bar{R}_w}{d_2} \quad (3.7)$$

$$\bar{R}_w = \frac{1}{m} \sum_{i=1}^m R_{wi}, \quad (3.8)$$

$$R_{wi} = \max(w_i) - \min(w_i), \quad (3.9)$$

where  $d_2$  is a group size dependent parameter and  $R_{wi}$  is the sample range of a given group  $i$ . This strategy allows to reset the properties (3.1) of the Shewhart  $\bar{X}$  control chart to

$$\text{UCL} = \bar{w} + k \frac{\bar{R}_w}{d_2 \sqrt{n}} \quad (3.10a)$$

$$\text{CL} = \bar{w} \quad (3.10b)$$

$$\text{LCL} = \bar{w} - k \frac{\bar{R}_w}{d_2 \sqrt{n}}. \quad (3.10c)$$

Charts based on sample standard deviation are called Shewhart  $S$  control

charts and consider

$$\text{UCL} = \bar{s}_w + k \frac{\bar{s}_w}{c_4} \sqrt{1 - c_4^2} \quad (3.11a)$$

$$\text{CL} = \bar{s}_w \quad (3.11b)$$

$$\text{LCL} = \bar{s}_w - k \frac{\bar{s}_w}{c_4} \sqrt{1 - c_4^2} \quad (3.11c)$$

following the aforementioned sample standard deviation approximations.

Finally, charts based on range are called Shewhart  $R$  control charts and are built considering

$$\text{UCL} = \bar{R}_w + k \frac{\bar{R}_w}{d_2} d_3 \quad (3.12a)$$

$$\text{CL} = \bar{R}_w \quad (3.12b)$$

$$\text{LCL} = \bar{R}_w - k \frac{\bar{R}_w}{d_2} d_3 \quad (3.12c)$$

where  $d_3$  is a group size dependent parameter.

**Cumulative sum control charts, CUSUM charts** Although they are not as intuitive and simple as the previous charts, CUSUM charts are more efficient in the detection of small shifts suffered by the process mean. Considering the mean  $\mu_{wi}$  of  $m$  samples of size  $n$ , the cumulative sum is calculated by

$$S_m = \sum_{i=1}^m (\mu_{wi} - \bar{w}). \quad (3.13)$$

A typical CUSUM control chart shows a variation in a random pattern centered around zero. By applying the V-mask procedure, it is possible to determine whether a process is in of control as illustrated in Figure 3.8. A V shape is superimposed on top of the latest point of the cumulative sum control chart and, if the previous points lie between the V shape, the process is considered to be in control. Otherwise, it is suspected to be out of control and further analysis must be performed (NIST and SEMATECH, 2012).

**Exponentially weighted moving average control charts, EWMA control charts**

The EWMA statistics is the mean of all prior data weighted exponentially,



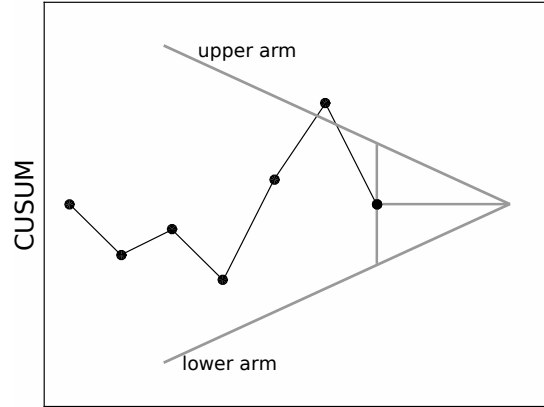


Figure 3.8: CUSUM control chart demonstrating an out of control process above the upper arm through the V-mask procedure (NIST and SEMATECH, 2012).

that is,

$$EWMA_i = \lambda w_i + (1 - \lambda) EWMA_{i-1}, \quad (3.14)$$

where  $w_i$  is the process variable value for instant  $i$ . The weighting factor  $\lambda$  determines the memory depth of the EWMA statistic according to the value it is set to (usually between 0.2 and 0.3). In opposition to the Shewhart control technique, the EWMA control procedure can sense a small drift depending on the choice of  $\lambda$ .

The control limits and the center line are

$$UCL = \overline{EWMA} + ks_{EWMA} \quad (3.15)$$

$$\text{Center Line} = \overline{EWMA} \quad (3.16)$$

$$LCL = \overline{EWMA} - ks_{EWMA} \quad (3.17)$$

where  $\overline{EWMA}$  is the target calculated based on historical data and  $s_{EWMA}$  is the standard deviation of the EWMA statistic obtained by

$$s_{EWMA} = \sqrt{\frac{\lambda}{2 - \lambda}} \sigma_w. \quad (3.18)$$

Nevertheless, process variables are oftentimes correlated and these univariate charts may produce false alarms. Multivariate statistical process control approaches are required in such situation.

A useful method to describe relationships between variables is based on Hotelling control charts. In these charts, the distance Hotelling's  $T^2$  is used to measure the covariance of a multivariate normal distribution defined by

$$T^2 = n (W - \mu)^\top \Sigma^{-1} (W - \mu) , \quad (3.19)$$

where  $n$  is the sample size,  $W - \mu$  is the deviation between observations and their mean, and  $\Sigma$  is the covariance matrix.

Also, the univariate exponentially weighted moving average control charts can be extended to the multivariate case (Yang and Sheu, 2006) in a model similar to (3.14),

$$EWMA_i = \Lambda W_i + (1 - \Lambda) EWMA_{i-1} , \quad (3.20)$$

where  $\Lambda$  is a diagonal matrix with weighting factors  $\lambda_i$  and  $W_i$  is a vector containing the process variables values for instant  $i$ . The multivariate EWMA control chart draws the Hotelling's statistic for the data  $EWMA_i$  by

$$T^2 = EWMA^\top \Sigma^{-1} EWMA . \quad (3.21)$$

A detailed review of multivariate control charts and their interpretation may be found in Cinar and Undey (1999) and Bersimis et al. (2007).

In spite of the usefulness of multivariate control charts, when the statistics exceed the upper control limit it is difficult to identify which variable generated the out of control signal and it is therefore advisable to run multivariate control charts in parallel with univariate control charts. Moreover, those univariate control charts alone will possibly not explain the out of control signal when this is caused by changes in variables covariance or correlation.

As referred to above, using multivariate control charts may be impractical for high dimensional system with collinear variables. A common strategy for tackling this problem is to apply projection methods such as principal component analysis (PCA), partial least squares (PLS) and support vector machines (SVM).

The principal components control chart, used to determine if the process is in control, is built based on the PCA method. It is worth emphasize that the usage of the principal components is advantageous since a small number of uncorrelated variables are able to capture most of the data variability.

PLS can also be used, condensing the problem to a set of latent variables that maximizes the covariance between the process and quality spaces.

Finally, SVM, a machine learning method for linear and non-linear rules, is used in the present context to optimize large classes of multivariate non-linear performance measures that are computationally tractable.

A description of these three methods may be found in Section 2.2.4.

### ■ MVC-based benchmarks

The minimum variance control benchmark (MVC), often called Harris index, is one of the most popular methods to determine controllers performance. For a given stable single input single output (SISO) process with a time delay  $\tau$ , the closed-loop relationship between the unmeasured disturbances  $w(t)$  and the process output  $y(t)$  can be expressed as an infinite order moving average process (Jelali, 2010), that is,

$$y(k) = w(k) \sum_{i=0}^{\infty} f_i z^{-i}, \quad (3.22)$$

where  $f_i$  are the Markov parameters,  $f_i z^{-i} = Z\{f(t - i\Delta t)\}$ , and  $\Delta t$  is the sampling time. Based on works of Åström (1970), Box et al. (1970) and DeVries and Wu (1978), Harris (1989) proposed to use the minimum variance (MV) controller as a lower bound to assess the performance of single loop controllers, which is estimated taking into account the process time delay by

$$\sigma_{\text{MV}}^2 = \sigma_w^2 \sum_{i=0}^{b-1} f_i^2, \quad (3.23)$$

where  $b$  is the integer number of sampling periods<sup>1</sup> correspondent to the process time delay  $\tau$  and calculated by  $b = 1 + \text{integer}(\tau/\Delta t)$ ,  $\sigma_w^2$  is the unmeasured disturbances variance, and  $\sigma_{\text{MV}}^2$  represents the output variance obtained by the application of the minimum variance controller to a time-series model estimated from measured output data.

In spite of its simplicity, the minimum variance controller can generate large input signals. Also, a closed-loop under MVC quite often has poor robustness

<sup>1</sup>For discrete systems with no time delay, the value of  $b$  is considered as 1 (one) because the actual output depends on the previous input (Jelali, 2010, page 27).

properties. These issues motivated the development of better performance indices (Eriksson and Isaksson, 1994). Desborough and Harris (1992) have adopted the closed-loop potential factor  $\eta_{\text{Harris}}$  defined by

$$\eta_{\text{Harris}} = \frac{\sigma_{\text{MV}}^2}{\sigma_y^2} = \frac{\sum_{i=0}^{b-1} f_i^2}{\sum_{i=0}^{\infty} f_i^2}. \quad (3.24)$$

where  $\sigma_y^2$  is the actual output variance extracted from measured data. This index may be easily estimated by the algorithm developed in Huang and Shah (1999). And the work of Desborough and Harris (1993) normalized the index resulting in

$$\eta_{\text{Harris}}^{\text{normalized}} = 1 - \frac{\sigma_{\text{MV}}^2}{\sigma_y^2} = 1 - \frac{\sum_{i=0}^{b-1} f_i^2}{\sum_{i=0}^{\infty} f_i^2}. \quad (3.25)$$

The estimation of Harris index is possible from routine operating data, with no need for any additional experiments, which represents the main advantage of the method (Jelali, 2006). The minimum variance concept was also adapted to feedback/feedforward control loops (Desborough and Harris, 1993; Stanfelj et al., 1993; Huang et al., 2000a), cascade control loops (Ko and Edgar, 2000) and non-minimum phase systems (Tyler and Morari, 1995). Furthermore, the index was extended to cases of varying setpoints, a branch followed by Perrier and Roche (1992) and Ko and Edgar (2000).

Some authors modified the Harris index (originally devoted to SISO systems only) to deal with multivariate systems (Huang and Shah, 1998; Huang et al., 2005). Such extension involved the calculation of a time delay matrix (also known as the interactor matrix). However, as referred by Jelali (2006), the interactor matrix is not easy neither to understand nor to calculate motivating the development of methods that avoid its use (Ettaleb, 1999; Ko and Edgar, 2001b; McNabb and Qin, 2003). For example, Ettaleb (1999) defined the performance index as

$$\eta_{\text{Harris}} = \frac{\sum_{i=1}^p \sigma_{i,\text{MV}}^2}{\sum_{i=1}^p \sigma_i^2}, \quad (3.26)$$

where  $p$  is the number of outputs of the MIMO process. Appendix B presents a summary of the modifications and extensions of the Harris index.

In order to quantify and diagnose the process variability, Farenzena (2008)

developed a new set of indexes that decompose the information contained in the Harris index into several components easing the task of its interpretation. Assuming a locally linear process, the controlled variable  $y(t)$  is decomposed into three components

$$y(t) = f(t) + g(t) + w(t), \quad (3.27)$$

where  $f(t)$  is the signal portion that may not be reached due to the time delay,  $g(t)$  the signal portion that may not be reached due to the feedback controller performance, and  $w(t)$  the white noise inserted in the process, which means that the total variance is given by

$$\sigma_{y(t)}^2 = \sigma_{f(t)}^2 + \sigma_{g(t)}^2 + \sigma_{w(t)}^2. \quad (3.28)$$

The researcher defined the three indexes

$$\text{deli} = \frac{\sigma_{f(t)}^2}{\sigma_{y(t)}^2}, \quad \text{tuni} = \frac{\sigma_{g(t)}^2}{\sigma_{y(t)}^2}, \quad \text{nosi} = \frac{\sigma_{w(t)}^2}{\sigma_{y(t)}^2}, \quad (3.29)$$

that quantify the time delay influence in the process variability, the impact of the feedback controller performance, and the white noise influence, respectively. If the influence of the time delay over the process is detected as too high, Farenzena (2008) recommends the application of the Smith Predictor or a feedforward technique to compensate the phenomenon. If the indexes analysis point that the feedback controller performance affects the process variability, the controller parameters should be retuned or a controller of higher order should be chosen.

The calculation of the nosi index is performed by the decomposition of the controlled variable: the signal  $\hat{y}(t)$  is fitted with an autoregressive model and the white noise component is quantified by applying the difference between the real signal and its prediction, that is,

$$w(t) = y(t) - \hat{y}(t). \quad (3.30)$$

Knowing the process time delay  $b$ , the component associated with the feedback controller is computed by a simple linear regression where dynamic data

are included by

$$g(t) = \underbrace{\begin{bmatrix} y(n-b) & y(n-b-1) & \cdots & y(n-b-m+1) \\ y(n-b-1) & y(n-b-2) & \cdots & y(n-b-m) \\ \vdots & \vdots & \ddots & \vdots \\ y(m) & y(m-1) & \cdots & y(1) \end{bmatrix}}_X \underbrace{\begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_m \end{bmatrix}}_\alpha, \quad (3.31)$$

where  $n$  is the number of samples,  $m$  is the autoregressive model order, and  $\alpha$  are the model parameters calculated by

$$\alpha = (X^\top X)^{-1} X^\top y', \quad (3.32)$$

with

$$y' = \begin{bmatrix} y(n) \\ y(n-1) \\ \vdots \\ y(m+b) \end{bmatrix}. \quad (3.33)$$

The process variability contribution  $f(t)$  is now computed by subtracting the previously calculated components

$$f(t) = \hat{y}(t) - g(t). \quad (3.34)$$

### ■ Advanced benchmarks

Advanced benchmarks deal with extensions of the minimum variance concept that need more information about the plant than just the time delay  $\tau$ . Two well-known approaches are the generalized minimum variance (GMV) and the linear-quadratic Gaussian (LQG). In spite of being more realistic than other known benchmarks, both need more information on controller performance (such as how much the output variance may be reduced without significantly affecting the controller output variance or if the actuator wear is a concern) (Jelali, 2010).

Proposed by Grimble (2002), GMV minimizes the weighted sum of the control error  $e(t)$  and of the manipulated variable  $u(t)$ . The GMV cost function to be

minimized is defined by

$$J_{\text{GMV}} = E\{\phi(t)^2\}, \quad (3.35)$$

$$\phi(t) = P_c e(t) + F_c u(t), \quad (3.36)$$

where  $\phi(t)$  is the squared generalized output signal,  $E\{\cdot\}$  is the expected value,  $P_c$  and  $F_c$  are weighting functions. The dynamic weightings must ensure the stability of the closed-loop system and cannot be chosen arbitrarily, which represent be a problem.

The benchmark LQG, proposed by Huang and Shah (1999), may be computed after having detected poor performance with the Harris index. This benchmark does not require a specific controller implementation running over the process and provides the performance bound for any linear controller in terms of the weighted input and output variances. It provides useful information about how far the control performance is from the best achievable performance with the same control effort. Mathematically, the objective function associated to this approach is defined as

$$J_{\text{LQG}} = \sigma_y^2 + \rho \sigma_u^2, \quad (3.37)$$

where  $\rho$  is the move suppression weight.

Table 3.2 summarizes the advanced performance benchmark works.

### ■ Data-driven approaches

Notwithstanding the attractiveness of the mentioned approaches, all of them require information that is often difficult to obtain. To circumvent this problem, some benchmarks are specified based on historical data from to a time period when the plant was perfectly tuned and optimized. In practice, the baseline performance is defined by the engineer via visual inspection of the trend graphs in a predefined time window. This approach is known by different denominations as Jelali (2006) refers: historical data benchmark (HIS), reference dataset benchmark or baselines.

Another simpler data-driven performance index was developed by Qin and Yu (2007) using the covariance and eigenvalue functions.

In addition, a method based on the prediction error was developed by Huang

Table 3.2: Advanced performance benchmark overview.

Work	Observations
Grimble (2002)	Extended the MVC to the more flexible GMV.
Majecki and Grimble (2004)	Extended the GMV to multivariable systems.
Grimble (2004)	Derived the GMV control law for nonlinear multivariable systems.
Haibo and Maying (2010)	Proposed a GMV for cascade control systems.
Grimble and Majecki (2004)	Described a new method for the cost function selection knowing the existing control structure. It is developed based on the GMV, but it is also applicable to some cases of the LQG.
Kammer et al. (1996)	Used non-parametric modelling in frequency domain to ascertain the optimality of an LQG controller based on the comparison of the optimal and the achieved cost function.
Huang and Shah (1999)	Proposed a LQG not requiring that a specific controller be implemented for the given process.
Kozub (2002)	Discussed some critical issues related to LQG.
Dai and Yang (2004)	Proposed a simpler method for obtaining the LQG based on the subspace identification approach.
Danesh Pour et al. (2009)	Improved the problem with the consistency of the noise variance estimation in the original model of LQG.



and Kadali (2008). These authors suggested the fitting of the closed-loop data to obtain the Markov coefficients which are useful in the determination of a scalar measure of the covariance matrix. This measure is then used to define a dimensionless index indicating the performance potential of the system.

### ■ Other methods

Besides the described indices, specialized metrics have been developed to include design specifications of the user leading to more realistic performance indices. Referred by Jelali (2006) as user-specified performance indices, these are generally defined as

$$\eta_{user} = \frac{J_{user}}{J_{act}}, \quad (3.38)$$

where  $J_{user}$  is the corresponding value of the user-specified performance measure.

Desborough and Harris (1992) and Thornhill et al. (1999) proposed the use of the extended horizon performance index (EHPI) with the general expression similar to  $\eta_{Harris}$  defined in (3.24) but calculated for a time interval larger than the time delay  $\tau$ . This extended horizon avoids the time consuming determination of time delay and regarding the prediction horizon as an engineering criterion (Jelali, 2010).

Liu and Gao (2011) developed a multi-objective user-specified benchmark problem. The linear matrix inequality region method is applied to solve the pole placement constraint. Finally, a cone complementarity linearisation algorithm is used to handle the resulting non-convex problem.

Benchmarks with a more restricted structure or model based for assessment of controllers performance were also developed and applied (Jelali, 2006). Eriksson and Isaksson (1994) and Ko and Edgar (1998) suggested the first approach with the optimal PID benchmark (OPID), where a lower bound of the variance was calculated by restricting the controller type to PID and allowing for more general disturbance models. The index takes the form

$$\eta_{OPID} = \frac{\sigma_{PID}^2}{\sigma_y^2}, \quad (3.39)$$

where

$$\sigma_{\text{PID}}^2 = \min_K \sigma_y^2 \quad (3.40)$$

and  $K$  are the PID controller constants.

Later, model-based approaches which could explicitly handle constraints and longtime delays were suggested by Ko and Edgar (2001a); Schafer and Cinar (2004); Julien et al. (2004). Fundamental questions related to these approaches are addressed in works of Patwardhan (1999) and Dumont et al. (2002), specially the issues of the source of poor control performance (as bad controller tuning or inaccurate modelling).

Appendix B contains a summary of the modifications and extensions of the indices here presented.

The described indices/techniques detect poor performances in processes in a general way. Complementary indices were developed to determine, during the detection of poor performance in a plant, the primary origin of the disturbances responsible for that poor performance, mainly categorized into oscillating, non-oscillating and non-stationary disturbances.

### 3.2.2 Poor performance diagnosis

Once the assessment of a given control loop ended up in the detection of its poor performance, it is essential to find its root cause. Figure 3.9 shows the work flow of this diagnosis stage, from the disturbances type to the underlying causes. As reported by Thornhill and Horch (2007), the key tasks of the diagnosis are the detection of one or more periodic oscillations, the detection of non-periodic disturbances and plant upsets, and the determination of the locations in the plant affected by those oscillations/disturbances together with their most likely root causes.

#### ■ Disturbance type diagnosis

Firstly, it is important to identify the type of disturbances present in the control loop: oscillating, non-oscillating or non-stationary.

**Oscillating disturbances:** Frequent in industrial processes, they are usually caused by aggressive controller tuning, presence of non-linearities (stiction, hys-

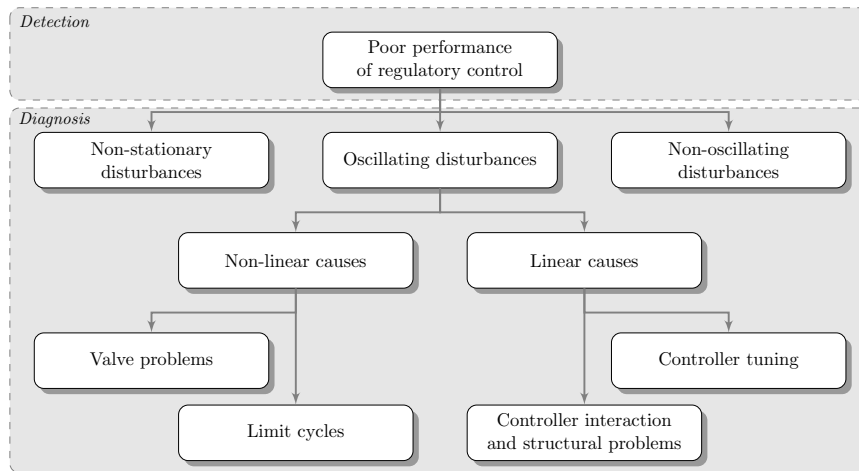


Figure 3.9: Poor performance detection and diagnosis.

teresis, etc) in control valves and internal/external disturbances. Due to the large percentage of systems with poor performance originated by problems/faults in control valves, the oscillating disturbances are the most studied ones (Desborough and Miller, 2002).

When oscillations are significant enough, they may be detected both in time and frequency domain. Surveys on the available methods were performed by Jelali (2006), Choudhury (2011) and Shardt et al. (2012). A classification for the existing methods is proposed in Figure 3.10.

The most widely used and described methods of the first group (time domain) are the settling time method (O'Connor and O'Dwyer, 2004), the integral of absolute error method (Hägglund, 1995; Hägglund, 2005), the autocorrelation and partial correlation functions method (Shumway and Stoffer, 2000), the poles of ARMA model method (Salsbury, 2006) and the zero crossings method (Thornhill and Hägglund, 1997). In the second group (frequency domain), the most popular methods are the Nyquist method, the Bode and Nichols plots and phase margins method (O'Connor and O'Dwyer, 2004; Huang and Shah, 1999), the maximum closed-loop log modulus method (Chiang and Yu, 1993; Belanger and Luyben, 1996), the autocorrelation method (Shumway and Stoffer, 2000) and the damping method (Miao and Seborg, 1999).

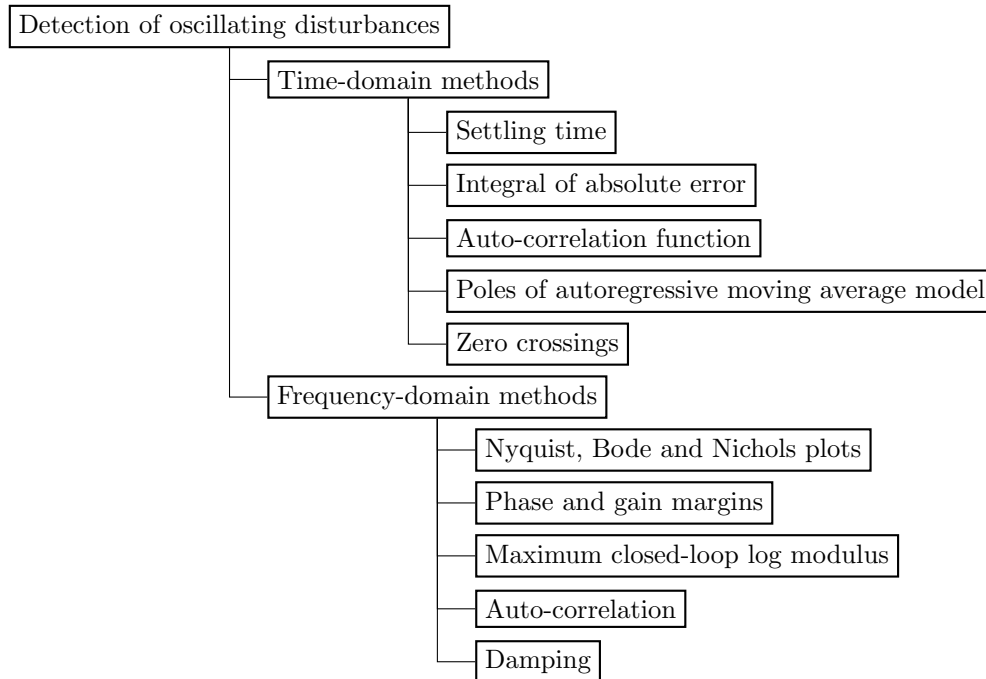


Figure 3.10: Methods for oscillating disturbances diagnosis.

Hägglund (1995) developed a method for detecting oscillations based on the monitoring of the integral absolute error (IAE) between consecutive zero crossings of the control error. In spite of its importance as it can determine oscillations of significant amplitude, the method needs the controlled variable to be centered on the reference variable and is very sensitive to noise, which constitute strong limitations to the method performance. Some enhancements were proposed by Thornhill and Hägglund (1997) in order to improve the real-time oscillation detection. The algorithm of the enhanced method consists of:

1. Choose the amplitude of oscillation,  $A_{lim}$ , and the admissible disturbances number,  $n_{lim}$ , to the considered supervision period.
2. Calculate the admissible integral absolute error of half-oscillation,  $IAE_{lim}$ , through

$$IAE_{lim} = \frac{2A_{lim}}{w_u}, \quad (3.41)$$

where  $w_u$  is the ultimate frequency approximated by  $w_u = 2\pi/T_I$ . If unavailable, the integral time  $T_I$  is also approximated by  $T_I = 2\Delta T_i$ , where  $\Delta T_i$  is the time between consecutive zero crossings of

the signal.

3. Monitor the IAE of half-oscillations using

$$\text{IAE}_k = \sum_{i=k_0}^k e_i, \quad (3.42)$$

since the time that the control error  $e$  changes its signal,  $k_0$ , until the latest sample,  $k$ .

4. Calculate  $\Delta T_i$  when two consecutive zero crossings are detected.
5. Update the  $\text{IAE}_{\text{lim}}$  based on the new  $\Delta T_i$ .
6. If  $\text{IAE}_k$  exceeds the  $\text{IAE}_{\text{lim}}$ , conclude that a disturbance has occurred and  $\text{load}_k = 1$ . Otherwise,  $\text{load}_k$  is held at 0.
7. Monitor the detected disturbances number,  $n_k$ , using the relationship

$$n_k = \gamma n_{k-1} + \text{load}_k, \quad (3.43)$$

where  $\gamma$  is a parameter related to the supervision period,  $T_{\text{sup}}$ , defined as  $\gamma = 1 - \Delta t/T_{\text{sup}}$  and  $\Delta t$  is the sampling time.

8. If  $n$  exceeds  $n_{\text{lim}}$ , conclude that the signal presents an oscillatory behaviour.

Assuming that the oscillation period and the IAE between zeros crossings of the signal might have little variability along the time, Forsman and Stattin (1999) introduced an index that expresses the regularity between the two quantities. The main strength of the method is its capacity of detecting asymmetric oscillations. Nonetheless, the method has also some drawbacks associated, namely the fact that the presence of noise affects significantly the method's performance, the method requires the controlled variable to be centered on the setpoint variable, and it detects oscillations independently of their significance. The correspondent algorithm is as follows:

1. Calculate the control error zero crossings  $t_i, i = 0, \dots, N$ .
2. Calculate the time intervals between consecutive zero crossings  $\delta_i$  and  $\epsilon_i$  by

$$\delta_i = t_{2i+1} - t_{2i} \quad \text{and} \quad \epsilon_i = t_{2i+2} - t_{2i+1}, \quad (3.44)$$

with  $i = 0, \dots, N/2$ . Figure 3.37 evidences the meaning of variables  $\delta_i$  and  $\epsilon_i$  (for  $i = 0$ ).

3. Calculate the IAE  $A$  e  $B$  by

$$A_i = \int_{t_{2i}}^{t_{2i+1}} |e_t| dt \quad \text{and} \quad B_i = \int_{t_{2i+1}}^{t_{2i+2}} |e_t| dt, \quad (3.45)$$

with  $i = 0, \dots, N/2$ .  $A_i$  and  $B_i$  (for  $i = 0$ ) are depicted in Figure 3.37.

4. Determine  $h_A$  and  $h_B$  so that

$$h_A = \#\{i < N/2; \quad \alpha < \frac{A_{i+1}}{A_i} < \frac{1}{\alpha} \wedge \gamma < \frac{\delta_{i+1}}{\delta_i} < \frac{1}{\gamma}\} \quad (3.46)$$

$$h_B = \#\{i < N/2; \quad \alpha < \frac{B_{i+1}}{B_i} < \frac{1}{\alpha} \wedge \gamma < \frac{\epsilon_{i+1}}{\epsilon_i} < \frac{1}{\gamma}\} \quad (3.47)$$

where  $\#S$  defines the number of elements in  $S$  and  $\alpha$  and  $\gamma$  are tuning parameters ( $\alpha, \gamma \in [0, 1]$ ).

5. Calculate the index  $h$  as

$$h = \frac{h_A + h_B}{N}. \quad (3.48)$$

6. Evaluate the index according to

- if  $h \approx 0.1$ : signal just affected by noise.
- if  $h > 0.4$ : oscillatory signal, it demands for a detailed analysis.
- if  $h > 0.8$ : very distinctive oscillatory pattern.

Miao and Seborg (1999) developed a method based on the decay ration of the auto-correlation function of the control error. The usage of this function is appealing because it reduces, by itself, the signal noise. However, the decay ratio of signals which contain multiple oscillations may lead to wrong conclusions. The algorithm is the following:

1. Calculate the auto-correlation function of the control error  $e$  (or of the controlled variable  $x$ ) by

$$\rho_{e,k} = \frac{\sum_{i=1}^{N-k} (e_i - \bar{e})(e_{i+k} - \bar{e})}{\sum_{i=1}^N (e_i - \bar{e})^2}, \quad (3.49)$$

where  $N$  is the number of points of the time series,  $\bar{e}$  is the control error mean of the sample of size  $N$  and  $k$  is the number of intervals corresponding to the time series delay,  $k = 1, \dots, N - k$ .

2. Determine the first two maxima and the first two minima values of  $\rho_e$ .

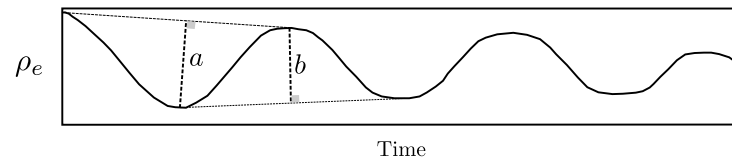


Figure 3.11: Variables meaning of Miao and Seborg (1999) method.

3. Calculate the line equation passing through the two maxima.
4. Calculate the line equation passing through the two minima.
5. Calculate the index  $R$  by

$$R = \frac{a}{b}. \quad (3.50)$$

The meanings of  $a$  and  $b$  are depicted in Figure 3.11.

6. Evaluate the index according to
  - if  $R \leq 0.5$ : control loop with acceptable performance.
  - if  $R > 0.5$ : control loop with severe oscillation.

Using also the auto-correlation function, Latwesen and Junk (2002) patented a method that, after visual detection of an oscillation, estimates the oscillation period based on the maxima and minima values of the function. This method reduces the error in the determination of the oscillation period even in signals containing overlaying of oscillations of different periods. Nevertheless, the need for visual detection and the characterization of exclusively the oscillation of smallest period reduces its applicability.

In the process of characterizing the oscillations present in a time series, Thornhill et al. (2003b) use the zero crossing of the auto-correlation function for determining the oscillation periods. However, the presence of multiple oscillations affects the determination of those zeros. Therefore, the method uses the determination of the signal spectrum to identify the dominant frequencies. Then, it applies a filter to remove the less significant oscillation periods. But the filtering process makes the automation of the method more difficult.

Nowadays, the overlay of oscillations with different periods still represents a challenge in what concerns the automatic detection of oscillations with no human interaction. With the objective of handling this kind of situations,

Srinivasan et al. (2007) and Srinivasan et al. (2012) proposed an approach using the modified empirical mode decomposition that is able to select the different signal oscillations. After the selection step, the zero crossings of the generated components are determined and the oscillation periods calculated.

Also, Wang et al. (2013) developed a method to characterize multiple oscillations present in a signal. This method makes use of the discrete cosine transform to convert the signal in its more elementary frequency components. The transform application makes the method computationally more intensive, though. Besides, the method requires the application of filtering as the noise strongly influences the method performance. The algorithm consists of:

1. Standardize the signal  $x$  by removing its mean and determine the respective discrete cosine transform  $z$ .
2. Apply the filter  $SL = 3\sigma_z$  using the function

$$z_{f,k} = \begin{cases} z_k, & \text{for } |z_k| \geq SL \\ 0, & \text{for } |z_k| < SL \end{cases} . \quad (3.51)$$

3. Select the segments  $[k_s, k_e]$  not null of transform so that

$$\begin{cases} z_{f,k_s} \neq 0 \text{ and } z_{f,k_s-r} = 0, & \text{for } r = 1, \\ z_{f,k_e} \neq 0 \text{ and } z_{f,k_e+r} = 0, & \text{for } r = 1, 2, 3, 4, \\ k_s \leq k_e . \end{cases} \quad (3.52)$$

4. Generate the inverse of the transform  $x'$  for each segment.
5. Obtain the zero crossing of  $x'$ ,  $t_i$ , and calculate the oscillation periods

$$T_i = t_{i+1} - t_i , \quad (3.53)$$

with  $i = 1, \dots, L$ , where  $L$  is the number of the found number of periods.

6. Calculate the regularity given by

$$r = \frac{\sqrt{\chi_{L-1, \alpha/2}^2} \bar{T}}{\sqrt{L-1} \ 3\sigma_T} . \quad (3.54)$$



7. If  $r > 1$ , the oscillation is considered regular and, consequently, the signal presents an oscillation of period  $\bar{T}$ .
8. Apply the filter  $SL = \sigma_z$  (similarly to step 2) to obtain a new  $z_f$ .
9. Repeat steps 3 to 7.
10. Select the  $\bar{T}$  (obtained through the filter application  $SL = \sigma_z$ ) with the higher indices  $F$ . The index  $F$  is defined by

$$F = 100 \left( 1 - \frac{\|x' - x\|_2}{\|x - \bar{x}\|_2} \right) , \quad (3.55)$$

where  $x'$  is the inverse transform of  $z_f$  corresponding to the period  $\bar{T}$ ,  $\bar{x}$  is the mean of signal  $x$  and  $\|\cdot\|_2$  represents the euclidean mean (dimension 2).

11. Localize the periods  $\bar{T}$  (obtained through the filter application  $SL = 3\sigma_z$ ) associated to the periods determined in the previous step.
12. Choose the dominant oscillations  $p$  with the criterion

$$p = \begin{cases} \bar{T}_{SL=3\sigma_z} & \text{if } r_{SL=3\sigma_z} \geq r_{SL=\sigma_z} \\ \bar{T}_{SL=\sigma_z} & \text{if } r_{SL=3\sigma_z} < r_{SL=\sigma_z} \end{cases} . \quad (3.56)$$

Although the detection of poor performance is a rather important task, the challenge is to trace the bad performance to its root causes. For this task, specialized methods and indices can be used. These indices do not require the knowledge of time delays nor any model identification. Instead, they are calculated based on the analysis of some measured signals, such as the manipulated, controlled and setpoint variables.

As already mentioned, there are different reasons for poor control performance, namely the limitations on achievable performance arising due to a combination of system and controller design, changes in system dynamics, varying disturbance, sensor or actuator faults, system non-linearities as well as other unknown sources (Jelali, 2006). The presence of some of these problems may render process dynamics non-linear (in opposition to the frequent assumption of linearity, at least locally, for the system). In this context, the root causes are usually divided into linear and non-linear (as indicated in Figure 3.9).

The root cause diagnosis is decomposed into two parts. In the first part,

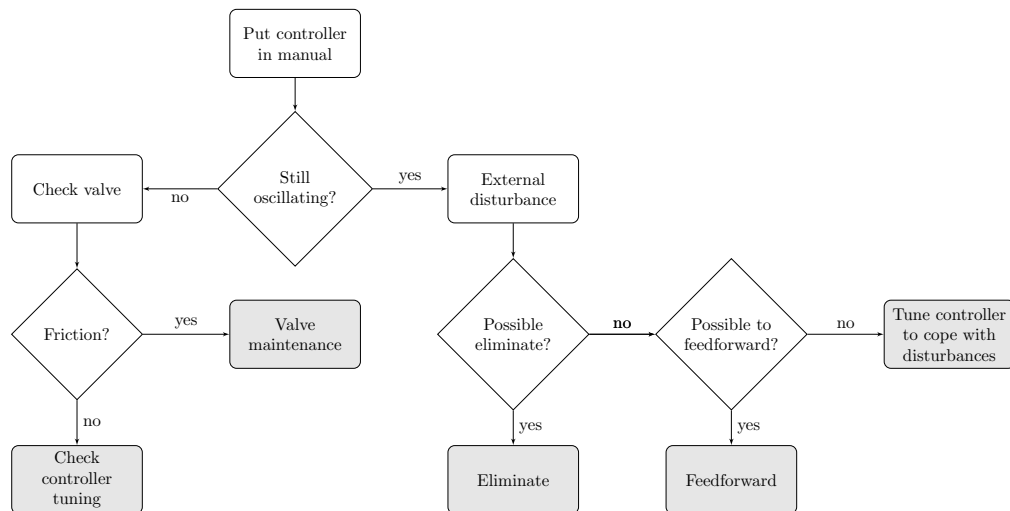


Figure 3.12: Manual diagnosis of oscillating disturbances (Hägglund, 1995).

the root cause of the disturbance is distinguished from other disturbances which will be posteriorly analyzed when the root cause is addressed. The second part is testing the candidate root cause loop to confirm the diagnosis report (Thornhill and Horch, 2007).

Hägglund (1995) proposed a procedure to manually diagnose oscillating disturbances represented in Figure 3.12), also described in detail by Huba et al. (2011). Apart from assuming that only stiction may be the oscillation cause, the method has a reduced vision, i.e., when the controller in a particular loop is set to manual (first step of the procedure) and the oscillation stops, its conclusion that the oscillation is caused by that control loop can be misleading: oscillations often arise from multivariable interactions between loops and the fact that the oscillation stops does not necessarily mean that that loop is where the root cause is (Huba et al., 2011). These two facts constitute the main disadvantages of the method.

The distinctive factor in the diagnosis of oscillating disturbances respects to the linearity/non-linearity of the signal generated by the disturbance. Common linear root causes of the disturbances comprise (Thornhill and Horch, 2007) poor controller tuning, controller interaction, and structural problems involving recycles while non-linear root causes of the disturbances include (Choudhury et al., 2008b) problems in control valves (such as stiction,

hysteresis, and deadband), on-off split-range control, sensor faults, process nonlinearities, hydrodynamic instabilities such as sluggish flows, and limit cycles.

**Non-oscillating and non-stationary disturbances:** Non-oscillating disturbances are generally characterized by their spectra which may have broad-band features or multiple peaks. Spectral decomposition methods are used to detect this kind of disturbances because they may distinguish significant spectral features from broad-band noise that spreads all across the spectrum (Thornhill et al., 2002; Xia and Howell, 2005; Xia et al., 2006; Tangirala et al., 2007). The spectral envelope method was also developed for detecting and categorizing process measurements with similar spectral characteristics (Jiang et al., 2006a). Another method for the diagnosis of non-oscillating random load changes based on a normalized index which is related to the damping ratio of a second-order model is described by Salsbury (2005).

In what concerns disturbances characterized by magnitude changes or that appear and disappear without a visible cause, designated as non-stationary disturbances, they can be diagnosed by the wavelet model developed by (Matsuo et al., 2003).

### ■ Specific malfunction diagnosis

Once the type of oscillation is determined, the diagnostic focuses in the task of finding the underlying causes. Two important root causes are PID controller associated problems and control valve related problems.

**PID controller problems:** PID controllers play a fundamental role in process industry keeping processes safe, stable and profitable. In order to reduce process variability, the controller must respond quick and appropriately to process load disturbances and setpoint changes. Moreover, it should coordinate its action with other controllers when belonging to control schemes such as cascade, ratio or feed-forward control.

In spite of the important functions they are supposed to perform, many of the control loops are not properly configured. Actually, around 75% of them may be increasing the variability of the process as reference variables are not

followed, control valves are oscillating, and some control loops are in manual control. Buckbee (2008) has reported a surprising set of statistics: 30% of control loops are improperly configured in the DCS, 85% of them have sub-optimal tuning and 15% of the control valves are improperly sized.

This poor performance is usually due to improper controller algorithm, improper tuning, over- or under-filtering, improper control loop configuration, and improper spanning. The use of an improper algorithm for the controller may lead to the inability to track the setpoint or to the creation of sustained oscillations. It also can generate excessive wear and tear of the control valve as well as excessive process movement in response to setpoint changes. In order to identify PID controllers with algorithm problems, a few general rules may be applied by the operator to a list of all the control loops in the plant suffering of poor performance. Buckbee (2008) suggests the following rules:

- Rule 1. Avoid derivative action on error because it causes a kick at setpoint changes and excessive wear of the valve. It is specially problematic when used on the inner loop of a cascade control scheme. Instead of derivative on error, it is recommended to use derivative of the controlled variable.
- Rule 2. Avoid gap control on the PID controller algorithm because it does not allow to track the setpoint closely leading to sustained oscillation or to an off-set between the controlled and the setpoint variables.
- Rule 3. Avoid integral action only when the controller under this algorithm does not respond quickly to load disturbance and setpoints changes.

The response of a controller is said to be sluggish when the process variable stays away from its setpoint for large periods of time. Poor tuning (improper controller constants) originates sluggish responses, deficiently handling the process upsets and ending up in propagating oscillations through the whole process. The usual operator response to these effects is to switch the controllers to manual mode disabling the loop action and compromising the safety and product quality.

In order to timely detect improper tuning some indices may be computed and monitored. The most used method implemented in vendor tools is the variance index (Xia and Howell, 2003; Zang and Howell, 2003). The Idle index was proposed by Hagglund (1999) to detect sluggish responses of controllers. Further improvements to deal with noisy data were proposed by Kuehl and Horch (2005). Howard and Cooper (2010) also proposed a new index to monitor the performance to disturbance rejection by applying a second-order under-damped model as pattern recognition.

An extreme case of poor performance due to improper tuning is when the process variables oscillate due to an aggressive controller. Ingimundarson (2006) developed an index based on the normalized partial derivative of the variance to deal with this situation. A positive synthetic gradient indicates a controller tuned aggressively while a negative gradient identifies a sluggish controller.

Poor performance can also be related to filtering. Filtering is a technique usually applied to reduce the impact of noise preventing an over-reaction of the controller to its presence in the feed signal. Since filters implementation may be performed in multiples places (in the instrument, in the controller software, and in the PID controller block), the lack of implementation consistency may lead to the application of more than one filter for a single loop and results in additional lags in the response. Such phenomenon is called over-filtering. Under-filtering, presenting the opposite effect, may also occur. Any excess of noise entering the control algorithm is amplified by the controller (specially with large controller gain or derivative action) inducing large process variability and compromising its quality and stability. For this reason, Buckbee (2008) suggests to keep all filtering in one place and choose the filter constant based on the process dynamics. Welander (2010) also proposes to coordinate simultaneously the determination of the filter constant and the derivative time.

Also, an improper configuration of the control loop may also be a cause for the decrease of its performance. Loops do not work in a vacuum and, consequently, their operation must be coordinated with the other control loops around. This is particularly important in the cases of cascade and ratio con-

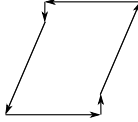
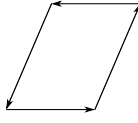
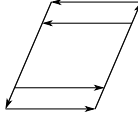
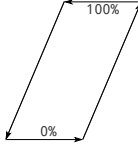
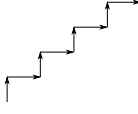
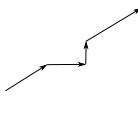
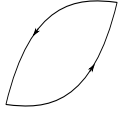
control strategies where the operation of a loop relies on other loops information. Poor performance induced from interaction or improper configuration may be diagnosed by the variance index (Xia and Howell, 2003), the causality method (Bauer et al., 2004, 2007), or multivariate analysis (Rossi et al., 2006). Anyway, interaction among loops, improper loop pairing and competing controllers are issues that still require more research attention.

Finally, an improper spanning (under- or over-spanning) may influence control loop performance as well. Usually, the under-spanned instrument identification may be performed by monitoring the metric defined by the percent of time at which the controlled variable exceeds the span limits.

**Valve problems:** Control valves are the most common final control element in chemical plants. They may contain different non-linearities such as stiction, backlash, deadband, saturation, and quantization (Bonavita et al., 2006), described succinctly in Table 3.3. From these non-linearities, stiction is the most common and one of the long-standing problems in process industry (Choudhury et al., 2005). Its detection is mainly performed by methods based on signals shape, surrogate analysis, and system identification (see Section 3.2.4 for their description).

The interest on other phenomena affecting control valves has not been much, maybe motivated by the fact that their consequences are not as prominent as those of stiction. Just recently, Xu et al. (2015) developed an approach to perform deadband online detection for a flow control valve based on a mathematical model. Relatively to backlash, Hägglund (2007) have proposed a method to detect the phenomenon in valves for stable processes. Techniques to quantify and compensate the valve problem were also proposed by the same author. Following this work, Haventon and Öberg (2008) suggested some method improvements in order to guarantee robustness and automation of the backlash estimation procedure. A second method for backlash detection was introduced by Ling et al. (2007). It uses a non-parametric statistical procedure to diagnose the phenomenon. Furthermore, the method identifies other valve malfunctions besides backlash, classifying the individual faults by extraction and analysis of the geometric features. Focusing on integrating processes, Farenzena and Trierweiler (2012) pro-

Table 3.3: Description of the most common non-linearities of the control valves (adapted from Bonavita et al. (2006)).

Nonlinearity	Definition	Representation of output vs. input
Stiction	Property of an element such that its smooth movement in response to a varying input is preceded by a static part followed by a sudden jump.	
Deadband	The range to reverse direction through which the input may be varied without initiating a response.	
Backlash	The play or loose motion in an instrument due to the clearance existing between mechanically contacting parts. Lost motion after reversing direction.	
Saturation	The controller requires more action than the actuator is able to deliver. May lead to wind-up related problems if not taken into account.	
Quantization	The process of transforming a continuous signal into one of finite steps or levels, as in an A/D converter.	
Deadzone	The range to keep the motion through which the input may be varied without initiating a response.	
Hysteresis	The path dependent characteristic attributed to materials not being able to return to their original shape and size after being stretched or deformed.	

posed the backlash index that distinguishes between stiction and backlash phenomena based on process variable patterns.

Another valve problem is that related to saturation that these elements can suffer (valves can not open or close beyond their physical range of  $[0, 100]\%$  even if the controller demands so). The possibility of valve saturation should

be checked when facing a situation of poor performance in the control loop. A simple method to perform this task is to monitor the manipulated variable movement when the controller is in automatic mode. It is noteworthy that sometimes saturation is deliberate, e.g., the controller will certainly lead some valves to the saturation when an operator wants to maximize the process throughput (Choudhury et al., 2008b).

### 3.2.3 Performance improvement

Once the poor performance diagnosis has been concluded, there is need to suggest improvement measures, namely the inspection and maintenance of the loop elements, the controller retuning, the controller redesign, and/or the compensation of valve problems, which should be selected depending on each particular situation.

When the poor performance origin lies on problems or malfunctions of components such as sensors and actuators, it is essential to perform inspection / maintenance work. For instance, if a control loop is identified as having an oscillating disturbance (derived from a valve suffering of stiction, for example), some valve tests must be carried out to confirm that is the real root-cause (valve inspection). If the suspicion is confirmed, the valve should be replaced (equipment maintenance) (Jelali, 2006).

As an example of a maintenance procedure, one can also refer the controller retuning (Veronesi and Visioli, 2010, 2015) by setting new values to the controller parameters. Section 4.2.1 describes important aspects and methods necessary to improve control loop performance by PID controllers tuning.

The controller redesign is another loop performance improvement alternative. The introduction of specialized procedures (as, for example, anti-windup, time-delay compensation, gain-scheduling or adaptive control) into the basic control strategy can enhance process control. Jelali (2006) argues that most of control performance problems are due to the lack of time-delay compensation and to the negligence of system interactions.

Although the solution for problematic valves is, as referred above, to perform maintenance work on the equipment, this is seldom possible in a running plant because of operation and safety considerations. Consequently, a problem-



atic valve may remain in operation for months until the next turnaround. A possible temporary solution to address these cases is the compensation of the phenomena occurring in the valve in order to eliminate or reduce their effects, to increase the valve lifetime, and to reduce the maintenance costs. Kato and Hatanaka (1998) invented a method for compensating backlash that computes the control signal based on the estimation of the disturbance acting on the system. Non-linear model predictive control is also an appealing compensation solution due to its ability to handle non-linearities and constraints. Several other applications to compensate valve problems were reported in literature. For example, Zabiri and Samyudia (2006) implemented a model predictive control strategy to control a plant under saturation and backlash. Also, Su et al. (2009) and Jang et al. (2005) studied the compensation in systems exhibiting both dead-zone and saturation. Rodríguez-Liñán and Heath (2012) developed an approach similar to that of Zabiri and Samyudia (2006) by including the inverse backlash into the model predictive control formulation which resulted in a set of mixed-integer inequalities. Stiction compensation methods are reviewed in Section 3.2.4.

### 3.2.4 Stiction modeling, detection/quantification and compensation<sup>2</sup>

Modern chemical plants consist of a large number of process units that have hundreds or thousands of control loops (Xu and Bao, 2010). These are essential assets because they ensure a high quality of the products as well as the safety of personnel and equipment (Alemohammad, 2011). Maintaining their performance is usually very time consuming (Yamashita, 2004) but necessary because of the increasing environmental, societal and competitive demands, aggravated by the lack of adequate training and experience of the staff in process control troubleshooting, as pointed out by Desborough and Miller (2002).

The main root causes for poor control performance are classified as non-stationary, non-oscillating and oscillating disturbances (Thornhill and Horch, 2007).

---

<sup>2</sup>This section is a reproduction in part with permission from Brásio, A. S. R., Romanenko, A., and Fernandes, N. C. P. (2014). Modeling, detection and quantification, and compensation of stiction in control loops: The state of the art. *Industrial & Engineering Chemistry Research*, 53(39):15020–15040. URL <http://dx.doi.org/10.1021/ie501342y>. Copyright 2015 American Chemical Society.

The later have received most of the research attention because they occur frequently in industrial processes (Desborough and Miller, 2002) due to linear and nonlinear phenomena. The linear causes comprise mainly aggressive controller tuning, interactions between controllers, and structural problems. In what concerns the nonlinear causes, they may arise in the process from phenomena such as valve malfunctions (Thornhill and Horch, 2007), namely hysteresis, backlash, deadband, and, especially, stiction. In fact, stiction is one of the long-standing problems in the process industry (Jelali and Huang, 2010) causing limit cycles and undermining economic performance of the assets.

Although the definitive solution for a sticky valve is to perform maintenance work on the equipment (Gerry and Ruel, 2001), this is seldom possible in a running plant because of operation and safety considerations. Consequently, a sticky valve may remain in operation for months until the next turnaround. The common industrial practice for compensating the stiction phenomenon is the manual detuning of the respective controller in order to eliminate or reduce the limit cycle effect. However, while the oscillation may decrease, the closed-loop performance of the process also deteriorates. The adequate mitigation of the stiction phenomenon allows to guarantee a high-level performance of the control loops, to extend control valves life time and to reduce maintenance costs.

While considerable progress has been made in stiction modeling, detection/quantification, and compensation, there are only a few surveys on these topics (Armstrong Hérouvry et al., 1994; Ordys et al., 2007; Choudhury et al., 2008a; Jelali, 2010; Arumugam and Panda, 2011). Moreover, these surveys do not cover a significant part of the models and methods already proposed by several researchers. The objective of this work is to provide a systematic taxonomy of the approaches covered in previous reviews, to describe significant contributions that were left out and to encompass recent developments, resulting in a comprehensive, contextualized, and updated state of the art in the field.

This section introduces the concept of stiction with a discussion on its influence in control loops, studies stiction phenomenon models reported in the literature, presents a thorough survey about approaches to detect/quantify stiction, followed by a summary of stiction compensation methods that are useful while valve maintenance or repair is not viable. An overview academic and commercial

software with stiction diagnosis capabilities is also depicted. Finally, a summary of the published techniques is provided.

## ■ Stiction in control valves

### Definition and characterization of stiction

According to the Merriam-Webster Dictionary (2012), the word *stiction* results from the contraction of STactic and frICTION and was first mentioned in a journal of aeronautics in 1946 to emphasize the difference between the static and the dynamic frictions.

In spite of the large number of works about static and dynamic frictions, only Choudhury et al. (2005) have tried to define formally such phenomenon and have proposed the only available description of the mechanism that causes stiction. These authors defined *stiction* as a “property of an element such that its smooth movement in response to a varying input is preceded by a sudden abrupt jump called the *slip-jump*. *Slip-jump* is expressed as a percentage of the output span. Its origin in a mechanical system is static friction which exceeds the friction during smooth movement”.

According to ISA (1995), the phenomenon is measured as the difference between the final and initial position values required to overcome static friction. For instance, 4% of *stiction* means that when the valve gets stuck, it will start the movement only after the difference between the control signal and the valve stem position exceeds 4%.

The typical behavior of a valve suffering from *stiction* may be observed in a phase plot (Figure 3.13), where it is possible to distinguish a sequence of four components (*deadband*, *stickband*, *slip-jump* and *moving phase*) that occur in the following process:

1. When the stem of a control valve arrives to a rest position or changes the direction, the valve sticks (see point A in Figure 3.13). While it does not overcome the frictional forces, the valve stem maintains the position (between points A and C) resulting in *deadband* (between A and B) and *stickband* (between B and C).
2. After overcoming the static friction, the valve stem converts the potential

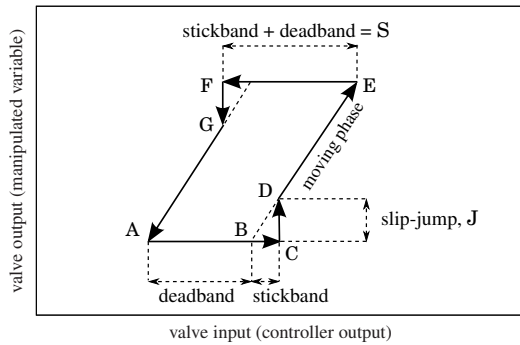


Figure 3.13: Phase plot showing the typical behavior of a sticky valve (Choudhury et al., 2008b).

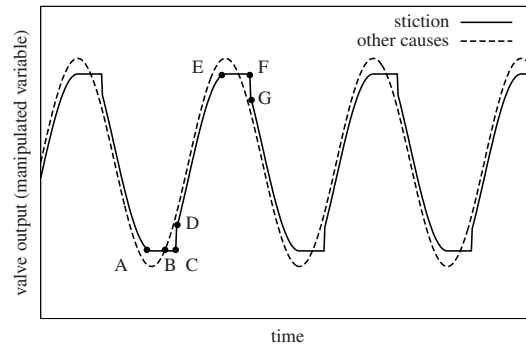


Figure 3.14: Distinction of limit cycles due to stiction and to other causes.

energy stored in the actuator into kinetic energy, jumping in an abrupt way to a new position. It is the *slip-jump* (between C and D).

3. Once the stem jumps, it continues to move until it eventually sticks again because of a stop or inversion of the direction of the stem movement (between D and E). This phase is called the *moving phase*.
4. During the *moving phase*, the valve stem may have a reduced velocity. This condition may stick the valve again while it keeps its travelling direction. In this case, there will be only *stickband* (the magnitude of the *deadband* is zero). This friction force is overcome if a valve input signal greater than the *stickband* magnitude is applied to the valve.

Tribology science classifies the described behavior in two regimes: the sliding regime and the pre-sliding regime (Altpeter, 1993; Swevers et al., 2000). The sliding regime occurs when there is a relative motion between two contacting surfaces being present in the *moving phase*. The pre-sliding regime occurs prior to the motion, when the external forces are compensated by the friction forces, and comprises the *deadband* and the *stickband*. The transition between the two regimes is done by the *slip-jump* mentioned above.

### Limit cycles due to stiction

If stiction is present, the behavior of a control loop deteriorates producing steady-state control errors or unwanted limit cycles in the valve stem position and, there-

fore, in the controlled variable (Armstrong H elouvry et al., 1994; Canudas de Wit et al., 1995; Olsson, 1996).

The limit cycles caused by stiction are characterized by distinctive wave shapes from those caused by other sources (Figure 3.14). The stem velocity of a valve exhibiting stiction remains at zero for a certain period of time, while other sources generate limit cycles behaving as sinusoidal waves (Choudhury et al., 2008b).

The fact that stiction induced limit cycles do not decay is an important issue because they cause permanent closed-loop performance degradation and undermine loop stability. An unstable behavior may appear when valve nonlinearities exceed nominal values. Using a Nyquist diagram, Srinivasan and Rengaswamy (2008) detected limit cycles in a control loop affected by stiction and highlighted the existence of an unstable limit cycle. The magnitude of stiction is a crucial element to determine the limit cycle behavior (Choudhury et al., 2004b).

Several effective techniques to analyze limit cycles behavior and to establish criteria of limit cycles stability in nonlinear systems have been proposed (Nayfeh and Mook, 1995; Somieski, 2001; Brito, 2011; Tsay, 2012). These techniques are not in the scope of this review and therefore will not be discussed.

### ■ Stiction modeling

Various approaches have been used to model the stiction phenomenon, although all of them represent a trade-off between the accuracy of the predictions and the simplicity of the model. Based on their shared characteristics, the models reviewed here were organized into the chart of Figure 3.15. The two major categories considered are the first-principle and the data-driven models (Garcia, 2008).

#### First-principle modeling

The first-principle models use the balance of forces and Newton's second law of motion to describe the friction phenomenon and belong to the following two classes: static or dynamic friction models (Olsson, 1996; Garcia, 2008).

**Static models:** The simplest models describe the friction force as a time-invariant function, using static functions of the stem velocity  $v$ . These models are often called static models.

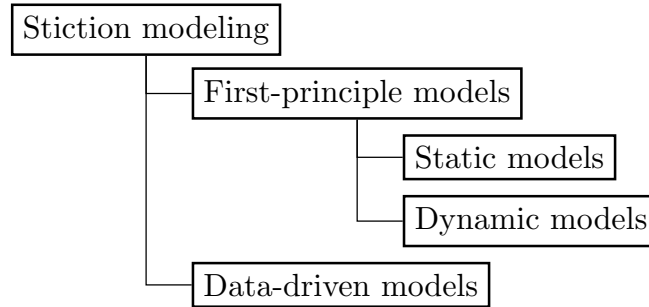


Figure 3.15: Compilation of the modeling approaches.

Models incorporating the classical friction components (Coulomb friction, viscous friction, and static friction forces) are summarized by Armstrong H elouvry et al. (1994) and Olsson (1996). The so-called Classical Model combines these classical friction components using the piecewise function (Garcia, 2008)

$$F_{\text{friction}} = \begin{cases} [F_C + (F_S - F_C) e^{-(v/v_S)^2}] \text{sign}(v) + F_V v, & \text{if } v \neq 0 \\ F_E, & \text{if } v = 0 \text{ and } |F_E| \leq F_S \\ F_S \text{sign}(F_E), & \text{if } v = 0 \text{ and } |F_E| > F_S \end{cases} \quad (3.57)$$

where  $F_C$ ,  $F_V$ , and  $F_S$  are the Coulomb friction, viscous friction and stiction coefficients,  $F_E = S_a P - kx$  is the external applied force,  $v_S$  is the Stribeck velocity,  $S_a$  is the diaphragm area,  $P$  is the air pressure,  $k$  is the spring constant, and  $x$  is the stem position. In this formulation,  $F_C$ ,  $F_V$ ,  $F_S$ ,  $S_a$ ,  $P$ , and  $k$  are unknown parameters. The values of  $S_a$ ,  $P$ , and  $k$  may often be defined by the valve specifications, while  $F_C$ ,  $F_V$ , and  $F_S$  are estimated model parameters. The Stribeck effect also described by the Classical Model is an effect found at very low velocities which consists of the continuous decreasing of the friction forces with increasing velocities. Figure 3.16 represents the friction force components and the Stribeck effect.

This model describes well the friction forces for steady-state velocities. However, numerical problems around zero velocity do not allow the prediction of complete stem stop typically exhibited by sticky valves. Karnopp (1985) proposed an approach to overcome the problems with zero velocity detection and, at the same time, to avoid switching between different equations for sticking and sliding. However, Sepehri et al. (1996) demonstrated that

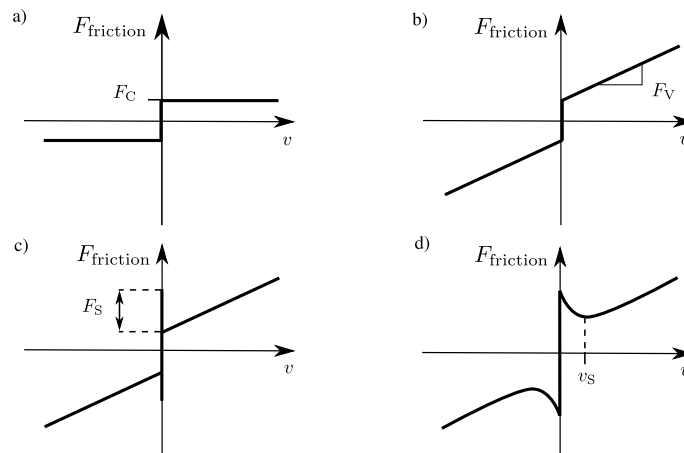


Figure 3.16: Friction force: a) Coulomb component; b) Coulomb and viscous components; c) Coulomb, viscous and static components; d) Coulomb, viscous and static components and the Stribeck effect. Adapted from Olsson (1996).

this model has numerical instabilities as well.

Later on, Leine et al. (1998) developed a modified version of the Karnopp Model trying to overcome its numerical problems using a set of ordinary non-stiff differential equations. Although this approach, designated by Switch Model, represented an improvement relatively to the previous ones, it still lacked the ability to describe friction thoroughly.

**Dynamic models:** From a control engineering point of view, it is preferable to consider stiction as a dynamic phenomenon (Åström, 1998). Armstrong Hérouvry et al. (1994) also pointed out the importance of including the effects of dynamic friction to complete the friction model.

Dahl (1968) explained the stiction phenomenon via an analogy with the stress-strain property of the materials. He proposed a model with the form (Olsson, 1996)

$$\frac{dF_{\text{friction}}}{dx} = \sigma_0 \cdot \left| 1 - \frac{F_{\text{friction}} \text{sign}(v)}{F_C} \right|^\alpha \cdot \text{sign} \left( 1 - \frac{F_{\text{friction}} \text{sign}(v)}{F_C} \right), \quad (3.58)$$

where  $\sigma_0$  is the stiffness coefficient, and  $\alpha$  is a parameter that determines

the shape of the stress-strain curve (usually set to 1). In the literature, this model is commonly simplified to Olsson (1996)

$$\frac{dF_{\text{friction}}}{dt} = \sigma_0 \cdot \left[ 1 - \frac{F_{\text{friction}}}{F_C} \text{sign}(v) \right] \cdot v. \quad (3.59)$$

Model parameters  $\sigma_0$  and  $F_C$  are estimated via empirical curve fitting of the experimental data. Although the Dahl Model represents well the Coulomb component in a dynamical form, it is unable to capture stiction and the Stribeck effect. In spite of this, as argued by Leonard and Krishnaprasad (1992), Dahl Model provides a realistic and reliable model of friction, specially during sinusoidal motions. After the Dahl Model appeared, the interest on dynamic friction models increased leading to further developments.

Within this research direction, the Seven Parameter Model, described by Armstrong H elouvry et al. (1994), is an empirical model that includes the pre-sliding and the sliding regimes. The friction force is defined as

$$F_{\text{friction}} = \begin{cases} \sigma_0 \cdot x, & \text{if } v = 0 \\ \left[ F_C + F_S(\gamma, t_d) \cdot \frac{1}{1 + \left( \frac{v(t - \tau_l)}{v_S} \right)^2} \right] \cdot \text{sign}(v) + F_V \cdot v, & \text{if } v \neq 0 \end{cases}, \quad (3.60)$$

with

$$F_S(\gamma, t_d) = F_{S,a} + (F_{S,\infty} - F_{S,a}) \frac{t_d}{t_d + \gamma}, \quad (3.61)$$

where  $x$  is the displacement,  $\gamma$  is the temporal parameter of the rising static friction,  $t_d$  is the time since becoming stuck also called dwell time,  $\tau_l$  is the time constant of frictional memory,  $F_{S,a}$  is the Stribeck friction at the end of the previous sliding period, and  $F_{S,\infty}$  is the magnitude of the Stribeck friction at the end of the previous sliding period. The typical values of the seven model parameters  $\sigma_0$ ,  $F_C$ ,  $\gamma$ ,  $\tau_l$ ,  $v_S$ ,  $F_V$ ,  $F_{S,\infty}$  are summarized by the authors. The sliding regime incorporates the Coulomb and the viscous friction forces, as well as the Stribeck effect. This model attempts to capture friction dynamics by introducing a time delay term. But since this is done exclusively in the sliding regime, the stiction phenomenon is oversimplified and the model does not capture the real behavior of the pre-sliding regime. Besides, it does not show a clear distinction between these regimes and, as



result, it fails to describe the transition behavior.

Another model in line with Dahl (1968) considerations has been proposed by Canudas de Wit et al. (1995). This model, presented as the Lund-Grenoble Model or, simply, as the LuGre Model, is described by

$$F_{\text{friction}} = \sigma_0 \cdot z + \sigma_1 \cdot \dot{z} + \sigma_2 \cdot v, \quad (3.62)$$

with

$$\dot{z} = v - \frac{|v|}{g(v)} z, \quad (3.63)$$

$$g(v) = \frac{1}{\sigma_0} [F_C + (F_S - F_C) \cdot e^{-(v/v_S)^2}], \quad (3.64)$$

where  $z$  is the average bristles deflection,  $\sigma_1$  is the micro-viscous damping coefficient,  $\sigma_2$  is the viscous damping coefficient, and  $g(v)$  is a function specifying how the average deflection depends on the relative velocity of the contacting surfaces. The values of the model parameters  $\sigma_0$ ,  $\sigma_1$ ,  $\sigma_2$ ,  $F_C$ ,  $F_S$ , and  $v_S$  were chosen according to the ranges summarized in the Armstrong H elouvry et al. (1994) work.

The LuGre Model employs the pre-sliding displacement (a component of the pre-sliding regime) as an averaged characteristic. Besides, it is able to account for several phenomena such as the Stribeck effect, hysteresis and stick-slip transitions. However, Olsson et al. (1998) found out that the model does not predict accurately some behaviors related to hysteresis. In studies by Hensen (2002) and Hensen et al. (2002), a frequency domain identification technique was applied to the first-principle LuGre Model in order to obtain the parameters related with the frictional pre-sliding behavior.

Friction forces in valves have been researched since long ago. For example, Rabinowicz (1951) studied the importance of the transition between pre-sliding and sliding and regarded friction as a function of displacement, observing a peak in the friction force for small displacements from the sticking point. Inspired by the Dahl Model, Bliman and Sorine (1995) developed a second-order linear dynamic friction model to describe this behavior. The

Bliman-Sorine Model uses a state-space formulation given by

$$\frac{dx_S}{ds} = A \cdot x_S + B \cdot u_S, \quad (3.65)$$

$$F_{\text{friction}} = C \cdot x_S, \quad (3.66)$$

with

$$A = -\frac{1}{\epsilon_f} \begin{bmatrix} 1/\eta & 0 \\ 0 & 1 \end{bmatrix}, \quad B = \frac{1}{\epsilon_f} \begin{bmatrix} f_1/\eta \\ -f_2 \end{bmatrix}, \quad \text{and} \quad C = \begin{bmatrix} 1 & 1 \end{bmatrix}, \quad (3.67)$$

where  $s$  is the space independent variable,  $x_S$  is the state variable vector,  $u_S = \text{sign}(v)$ ,  $\epsilon_f$  is a distance,  $\eta$  is a dimensionless model parameter, and  $f_1$  and  $f_2$  are forces. The authors identified the model parameters  $\epsilon_f$ ,  $\eta$ ,  $f_1$ , and  $f_2$  analytically.

Their model is viewed as two Dahl models connected in parallel, one with a fast dynamics and the other with a slow dynamics. The fast model intends to describe an highest steady-state friction. Subtracting the slow model from the fast model results in a peak that corresponds to the friction force. However, according to Olsson et al. (1998), this attempt was not very successful, because the resulting model is less efficient than the LuGre Model.

Dupont et al. (2000) analyzed and discussed the stiction and the pre-sliding displacement phenomena. They concluded that both dynamic Dahl and LuGre models are able to describe pre-sliding displacement but not stiction. Based on simulation studies, the authors derived an elasto-plastic dynamic model which takes into account both phenomena. The Elasto-Plastic Model is given by

$$F_{\text{friction}} = \sigma_0 \cdot z + \sigma_1 \cdot \dot{z} + \sigma_2 \cdot v, \quad (3.68)$$

with

$$\dot{z} = v \cdot \left[ 1 - \alpha(z, v) \cdot \frac{\sigma_0}{f_{\text{ss}}(v)} \cdot \text{sign}(v) \cdot z \right]^i, \quad (3.69)$$

and

$$\alpha(z, v) = \begin{cases} 0, & \text{if } |z| \leq z_{\text{ba}} \\ 0 < \alpha < 1, & \text{if } z_{\text{ba}} < |z| < z_{\text{max}} \\ 1, & \text{if } |z| \geq z_{\text{max}} \end{cases}, \quad (3.70)$$

where  $\alpha(z, v)$  is used to achieve stiction behavior (requirements on the the choice of  $\alpha$  are developed by the authors),  $i$  is an integer exponent used to govern the transition rate of  $z$  in order to achieve a better experimental match (typically  $i = 1$ ),  $f_{ss}(v) = \sigma_0 \cdot g(v)$  is related to the Stribeck friction curve defined by the LuGre Model,  $z_{ba}$  is the breakaway displacement, and  $z_{max}$  is the maximum presliding displacement.

This model was tested only by simulation using model parameters available in literature.

Swevers et al. (2000) proposed a more complex model, known as Leuven Model as

$$F_{\text{friction}} = F_h(z) + \sigma_1 \cdot \dot{z} + \sigma_2 \cdot v, \quad (3.71)$$

with

$$F_h(z) = F_b + F_d(z), \quad (3.72)$$

$$S(v) = F_C + (F_S - F_C) \cdot e^{-(v/v_S)^\delta}, \quad (3.73)$$

$$\dot{z} = v \left[ 1 - \text{sign}\left(\frac{F_d(z)}{S(v) - F_b}\right) \cdot \left|\frac{F_d(z)}{S(v) - F_b}\right|^n \right], \quad (3.74)$$

where  $F_h(z)$  defines transition curves modeling the hysteresis friction force,  $F_b$  is the value of  $F_h(z)$  at the beginning of a transition curve (i.e.,  $z = 0$ ),  $F_d(z)$  is the transition curve,  $S(v)$  is a function that models the constant velocity behavior,  $\delta$  is a parameter that depends on the geometry of the application (usually between 0.5 and 1), and  $n$  is a coefficient determining the transition curve shape.

The parameter values are identified in two phases. The first phase identifies the parameters that define the sliding regime ( $F_S$ ,  $F_C$ ,  $v_S$ ,  $\delta$  and  $\sigma_2$ ) based on data obtained from constant velocity tests over a velocity range. These parameters are estimated using a Markov estimator (weighted least squares). The second phase determines the parameters that define the pre-sliding regime (transition curve equation,  $F_d(z)$ , and  $\sigma_1$ ). These parameters are estimated based on experimental data where the applied force is slowly ramped up and down and on a maximum likelihood estimator.

This formulation allowed an accurate modeling in both sliding and pre-

sliding regimes without a switching function. It considers that hysteresis occurs for the non-periodic pre-sliding regime what represents an improvement of the model accuracy. In spite of the big improvement introduced by this hybrid hysteresis model with non-local memory, its use is limited to control design and analysis because of the associated implementation difficulties and a discontinuity in friction force function for some cases.

The Leuven Model was later modified by other authors. In particular, Lampaert et al. (2002) implemented the hysteresis force using the more efficient Maxwell slip model which allows to eliminate a stack overflow problem. They also modified the model to overcome a discontinuity in the friction force. Dupont et al. (2002) improved the Leuven Model to non-physical drift phenomena, which arise when the force is characterized by small vibrations below the static friction limit.

Later, Lampaert et al. (2003) presented a novel friction model called Generalized Maxwell-Slip (GMS) Model with the following form:

$$F_{\text{friction}}(t) = \sum_{i=1}^N F_i(t) + \sigma_2 \cdot v(t), \quad (3.75)$$

where  $F_i(t)$  are the elementary friction forces modeled as

$$\frac{dF_i}{dt} = k_i \cdot v, \quad (3.76)$$

for sticking (until  $F_i > \alpha_i \cdot S(v)$ ), and

$$\frac{dF_i}{dt} = \text{sign}(v) \cdot F_C \cdot \left( \alpha_i - \frac{F_i}{S(v)} \right), \quad (3.77)$$

for slipping (until velocity goes through zero). The parameters  $k_i$  and  $\alpha_i$  define the shape of the hysteresis curve and the Stribeck effect, respectively.

The model is compared by its authors with well-known existing models showing the capability of capturing accurately the major effects, such as the pre-sliding regime and the Stribeck effect and found to be appropriate for control purposes.

Since some of the parameters coincide with the previous models (the case of  $F_C$ ,  $F_S$ ,  $v_S$ ,  $\delta$  and  $\sigma_2$ ), their values were adopted in this formulation. In addition, a curve fitting of the hysteresis curve estimates the values of  $k_i$  and  $\alpha_i$ . Despite the model novelty, Jamaludin et al. (2009) identified that the GMS Model is complex and has a large number of parameters which complicates its use.

Recently, Ferretti et al. (2004) developed the single- and multi-state integral friction models based on Dahl Model and on modifications by Lampaert et al. (2002). This approach is advantageous in terms of computational efficiency and accuracy, two very important aspects in the compensation of friction. According to Makkar (2006), the assumption that the friction coefficient is constant with sliding speed and have a singularity at the onset of slip limits the applicability of these models.

Most of the aforementioned approaches use piecewise continuous friction models and that may be problematic from a numerical standpoint. Makkar et al. (2005) developed a simple continuously differentiable model that captures the major effects reported in the friction forces modeling. They proposed an empirical model that accounts for the Coulomb friction, the viscous and the stiction friction forces, as well as for the Stribeck effect. The model is described as follows

$$F_{\text{friction}}(v) = \gamma_1 \cdot \left[ \tanh(\gamma_2 \cdot v) - \tanh(\gamma_3 \cdot v) \right] + \gamma_4 \cdot \tanh(\gamma_5 \cdot v) + \gamma_6 \cdot v, \quad (3.78)$$

where  $\gamma_i$  denotes unknown positive constants that are varied manually in order to capture the enumerated forces and effects. The lack of matching experimental data to the analytical model reduces its usefulness in new formulations.

In order to simplify the determination of the model parameters for adaptive control algorithms, Márton and Lantos (2007) suggested a novel approach that clearly distinguishes the low and the high velocity regimes.

He and Wang (2010) presented a comparison between some well-established first-principle and data-driven models (models discussed in Section 3.2.4). Based on a thorough analysis and on the effectiveness in simulating valve

stiction, they proposed a semi-physical model to reproduce the first-principle model predictions with simpler numerical implementation. The model has the form

$$u_v(t) = \begin{cases} u_v(t-1) + K[e(t) - \text{sign}(e(t)) \cdot f_D], & \text{if } |e(t)| > f_S \\ u_v(t-1), & \text{if } |e(t)| \leq f_S \end{cases}, \quad (3.79)$$

where  $e(t) = u(t) - u_v(t-1)$ ,  $u_v(t)$  is the stem position (system output),  $u(t)$  is the actuator air pressure (system input),  $K$  accounts for the overshoot observed in the physical model,  $f_D$  is the static friction force, and  $f_S$  is the Coulomb/dynamic friction force. Because they derived the model from first principles, the authors used the parameters available in other similar approaches to obtain values for the three model parameters ( $K$ ,  $f_S$  and  $f_D$ ). The main drawbacks of this model are: the inability to describe the Stribeck effect and the fact that it consists of a piecewise function.

Liang et al. (2012) introduced a model totally based on the physical structures and conditions. Built upon a modeling and simulation platform specifically developed for hydraulic and mechanical systems, they simulated different nonlinear valve faults (stiction, deadband, leakage and saturation) and verified the model using some nonlinearity assessment measures in these simulations. The formulation has several parameters: the poppet diameter, the moving parts mass, the hole diameter, the spring constant, the diaphragm area, and the differential pressure in the valve. For simulation purposes, four major parameters were introduced: the Coulomb friction force, the stiction force, the clearance on diameter, and the higher displacement limit.

Tang et al. (2015) developed a new semi-physical model to describe stiction phenomenon based on a careful analysis using three signal conversion processes and takes into account backlash component. In opposition to previous models, the new model showed consistency with experimental results.

### Data-driven modeling

A detailed first-principle stiction model requires the knowledge of several parameters that are difficult to estimate. In addition, computational implementations of such models may be too slow for application purposes. Since data-driven modeling approaches overcome these two disadvantages, several works in this direction have been reported the literature.

However, such kind of models also present some drawbacks. In fact, as pointed out by Garcia (2008) and He and Wang (2010), they cannot fully capture the dynamics of the valve. Moreover, different models are based on different assumptions and the choice of the best approach for a specific physical valve is not clear (He and Wang, 2010).

Stenman's, Choudhury's, Kano's and He's models are the most recent and representative of the data-driven models. The proposal of Stenman et al. (2003) attempts to reproduce the jump of the valve stem after the stickband, phenomenon represented through an only parameter,  $d$ . The Stenman Model has the form

$$x_k = \begin{cases} x_{k-1}, & \text{if } |u_k - x_{k-1}| \leq d \\ u_k, & \text{if otherwise} \end{cases}, \quad (3.80)$$

where  $u$  is the controller output, and  $x$  is the real position of the control valve. For simulated and real data, the authors determined the model parameter via a local tree search approach using the likelihood as the optimality criterion.

Nevertheless, Choudhury et al. (2008b) showed that the predicted and observed behaviors of the Stenman Model do not match in the case of a sticky valve excited with a sinusoidal input. Choudhury et al. (2004a, 2005, 2006b) suggested a different version of the model aiming to improve the representation of the stiction phenomenon (Figure 3.17). Based on a thorough discussion of the term "stiction", they proposed to distinguish stiction from other valve nonlinearities. Therefore, their model contains two parameters: the amplitude of deadband plus stickband,  $S$ , and the amplitude of the jump after the static friction is overcome,  $J$ . Both parameters were manually set to simulate open- and closed-loops without stiction and with various magnitudes of stiction.

Since the Choudhury Model was able to deal only with deterministic signals,

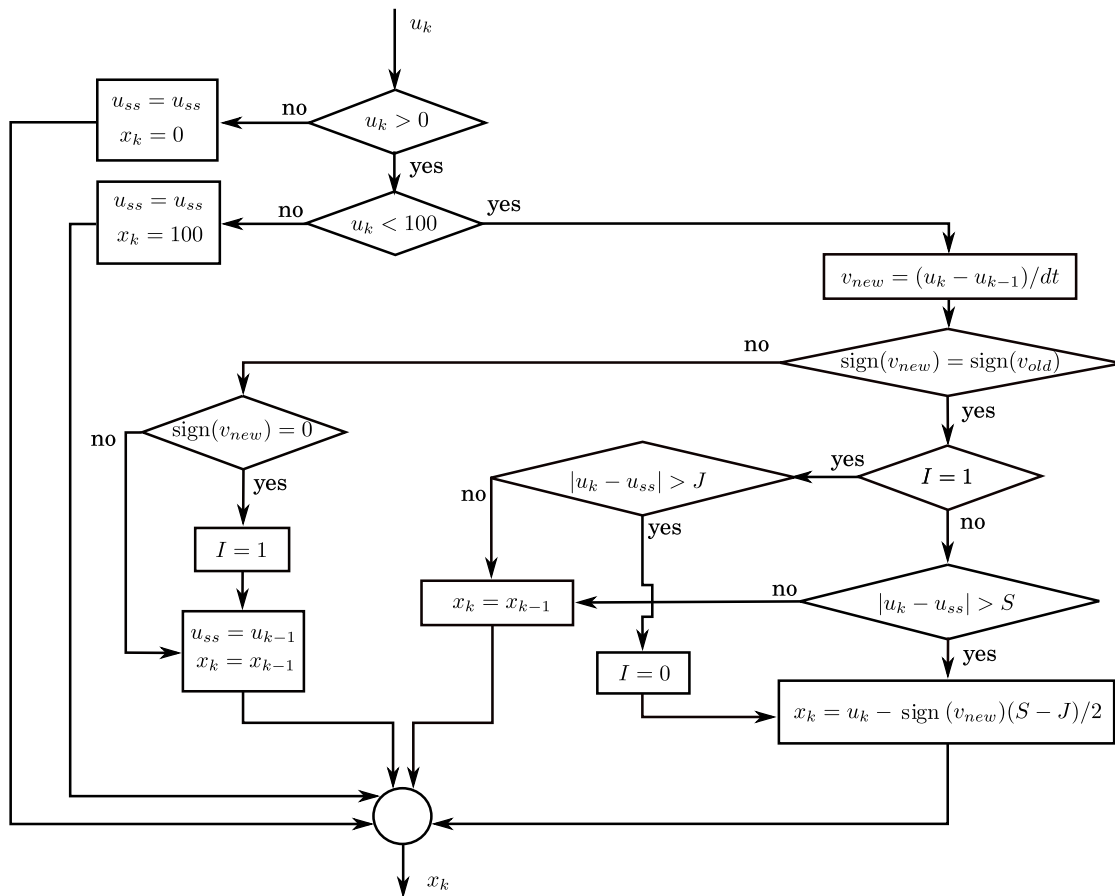


Figure 3.17: Choudhury Model (Choudhury et al., 2004a, 2005, 2006b).

Kano et al. (2004) and Maruta et al. (2005) developed a modified version that can handle broader situations (Figure 3.18). Trying to relate the parameters of the Choudhury Model with the elastic force, air pressure and frictional force, the authors redefined the two parameters. In the Kano Model,  $S$  corresponds to the sum of the static and dynamic frictions and  $J$  to the difference between the static and the dynamic frictions. Quantitatively, these parameters are equivalent to those of the Choudhury Model. Jelali and Huang (2010) compared these two models and concluded that both are able to predict satisfactorily the stiction effects.

He et al. (2007) developed a model that reduces the complexity of Kano’s and Choudhury’s formulations. Despite the structural simplification, the He Model presented in Figure 3.19 also uses two parameters. Besides, it has a more straightforward logic, naturally handles stochastic noise and reproduces industrial cases behavior (He and Wang, 2010). The model uses static  $f_S$  and dynamic  $f_D$  fric-



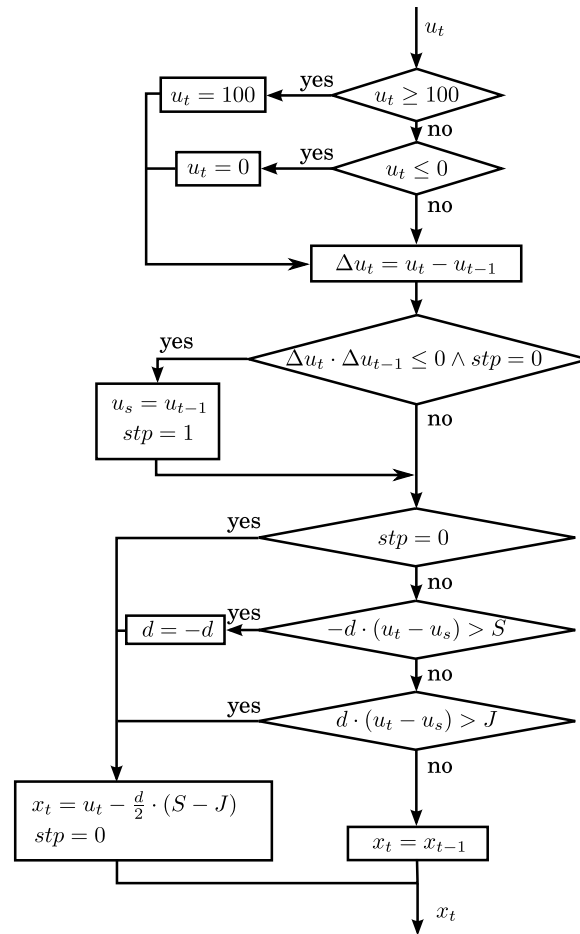


Figure 3.18: Kano Model (Kano et al., 2004).

tion parameters brought closer to the first-principle based formulation. To reduce the complexity, it uses a temporary variable representing the accumulated static force.

Other data based approaches have also been presented in the literature (Chen et al., 2008; Zabiri and Mazuki, 2010; Karthiga and Kalaivani, 2012).

The He Model assumes that the static friction is associated with all valve movement. This model requires that the static friction must be exclusively accounted for the pre-sliding regime. Chen et al. (2008) and Chen (2009) generalized the He Model to eliminate this disadvantage by introducing a two-layer binary tree logic (see Figure 3.20) that uses the two parameters defined by He et al. (2007). Although two extra variables are added to the He Model (the valve status flag,  $stop$ , and the movement direction,  $d_t$ ), the approach generalized the

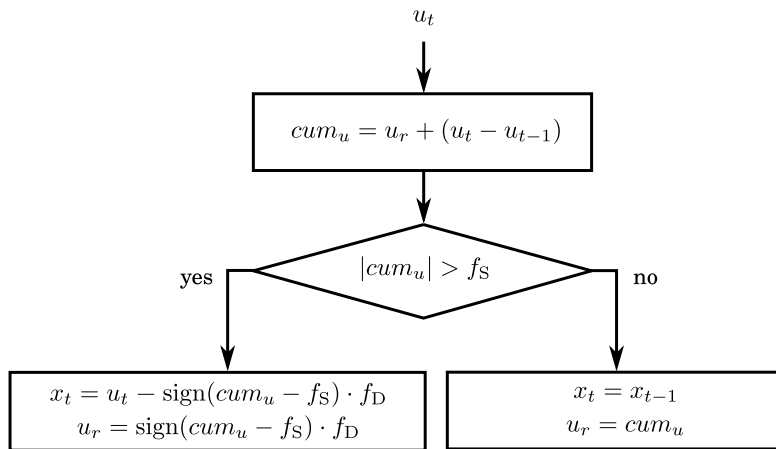


Figure 3.19: He Model (He et al., 2007).

static and the dynamic frictions improving the capture of various types of stiction patterns. Some simulations in open- and closed-loop showed the accuracy and the effectiveness of such stiction model.

Zabiri and Mazuki (2010) developed a black-box modeling approach based on a nonlinear autoregressive with exogenous input series parallel neural network. Numerical evaluations showed accurate predictions, even in multi-step ahead scenario. However, the model is robust only when stiction is less than 6% of the valve travel span.

Wang et al. (2010) proposed a blind approach to identify the system dynamics containing a sticky valve. Without an explicit parametrization of the nonlinearity, they used a Hammerstein Model to identify the system and disregarded the error propagation. The method was capable of capturing both the nonlinearity and the system dynamics in a feedback loop with a sticky control valve. However, this approach has a larger number of parameters to be estimated in comparison to other approaches considered in this section.

More recently, Karthiga and Kalaivani (2012) developed a new nonlinear data-driven model (Figure 3.21) considering three parameters: the deadband,  $d$ , the maximum pressure required to move the stem,  $u_{max}$ , and the stick-slip magnitude,  $f$ .

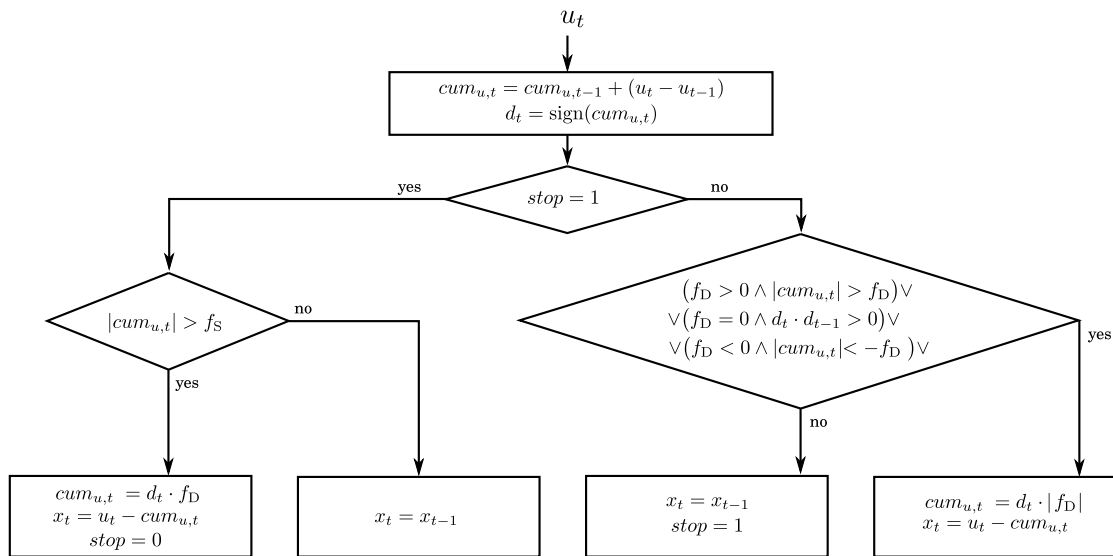


Figure 3.20: Chen Model (Chen et al., 2008; Chen, 2009).

### ■ Stiction detection /quantification

About 20 to 30% of all process control loops have oscillations due to stiction, resulting in losses of productivity (Kvam, 2009; Nallasivam et al., 2010). Because stiction is one of the major causes for oscillations, numerous techniques for its detection in linear control loops have been developed. These techniques take advantage of the nonlinearity introduced by stiction in the control valves in order to detect the presence of the phenomenon (Babji et al., 2012). Figure 3.22 shows the classification of existing methods for stiction detection and quantification and the relations among them.

### Shape-based methods

The first detection attempt was made by Horch (1999), through a simple method to diagnose oscillations in the process control loops. This pattern classification method is based on the cross-correlation between the controller input  $u$  and the process output  $y$  signals. The cross-correlation for the stationary signals  $u$  and  $y$  is obtained from

$$r_{uy}(\tau) = \sum_{k=0}^{N-|\tau|-1} u(k) y(k + \tau), \quad (3.81)$$

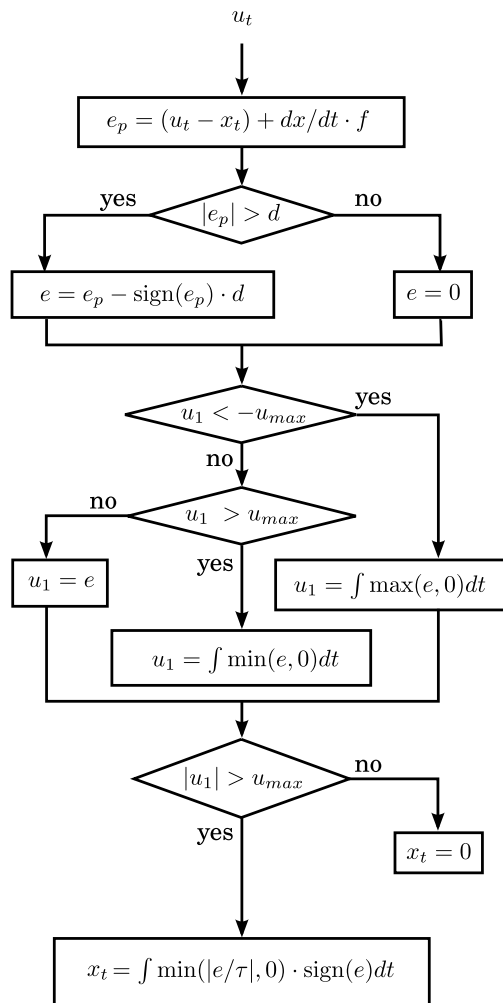


Figure 3.21: Karthiga Model (Karthiga and Kalaivani, 2012).

where the available datasets  $u$  and  $y$  are of finite length  $N$ , and  $\tau$  represents the lag. It allows to distinguish between the two most important causes of oscillations, stiction and external disturbances or unstable loop, and it is applicable to non-integrating processes controlled by PI controllers (Figure 3.23). As stated by Maruta et al. (2005), the main disadvantage of this method is that it is applicable only to systems with periodical fluctuations.

Later, Horch (2000) developed a method applicable to integrating processes using the probability density function (normalized raw histogram) of the second derivative of the process output. Basically, he compared two theoretical probability density functions characteristic of the stiction and non-stiction cases with the process output probability density function. The best fit determines whether

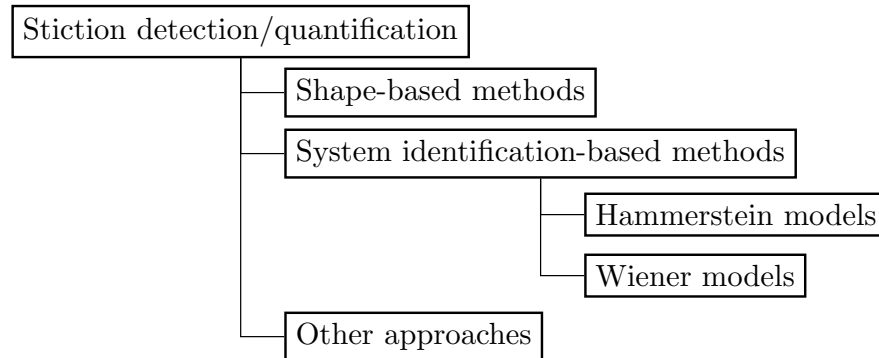


Figure 3.22: Compilation of the detection / quantification methods.

stiction is present (Figure 3.24).

Similarly, he applied the probability density function of the first derivative of the error signal for self-regulating processes. These methods required not only the data of the controlled variable, but also needed to know whether the process is an integrating or a self-regulating process.

Kano et al. (2004) explored the relationship between the valve input and the valve output and fitted a parallelogram to the phase plot. However, since the valve output is frequently impossible to measure, it is substituted by the controlled variable. This substitution changes the shape of the plot into an ellipse, making it difficult to be recognized by the algorithm because of the parallelogram shape assumption. In order to overcome the disadvantages of this method, Choudhury et al. (2006b,a) quantified stiction by fitting an ellipse to the valve input versus controlled variable data, where the maximum width of the ellipse was designated the apparent stiction (Figure 3.25). Based on the Choudhury et al. (2006b) method, de Souza L. Cuadros et al. (2010) proposed an improved algorithm to quantify stiction that selects the most significant points of the valve input and the controlled variable datasets and fits an ellipse. Although it has the disadvantage of being applicable to parallelogram patterns only (such as those generated by flow control loops), the authors argued that this new procedure estimates stiction with more precision than the Choudhury et al. (2006b) method for the considered patterns.

Choudhury et al. (2004a) proposed an automatic method to detect nonlinearities (such as stiction and backlash) that involves the calculation of higher-order

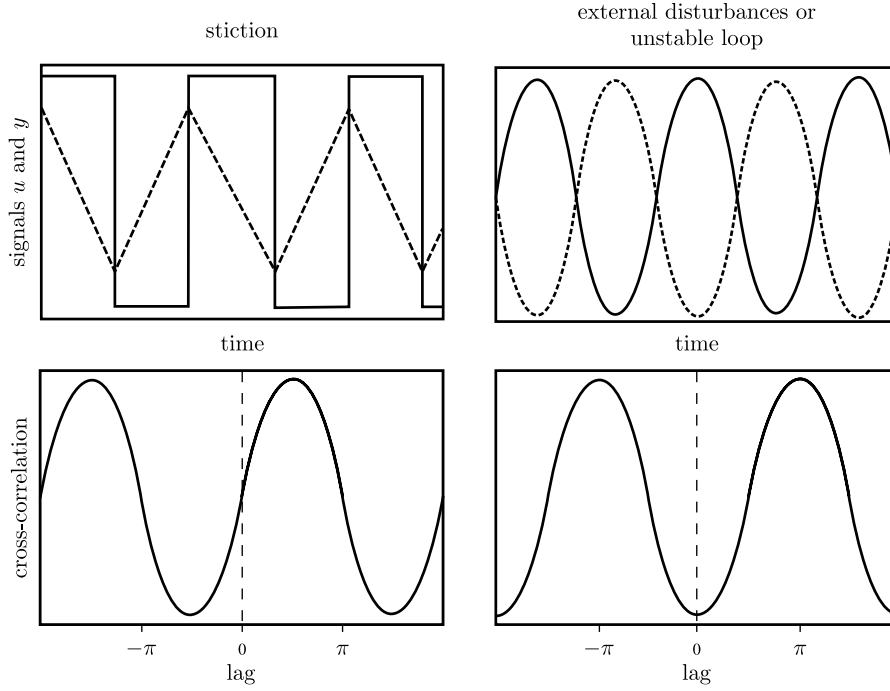


Figure 3.23: Horch method application on signals  $u$  (dashed line) and  $y$  (solid line). Oscillations due to stiction present an odd cross-correlation function, and oscillations due to external disturbances or unstable loop present an even cross-correlation function. Adapted from Horch (1999).

statistics of the closed-loop data. The statistical measures cumulants, bispectrum and bi-coherence of the control error signal are used to infer two metrics: the non-Gaussianity index,  $NGI$ , and the nonlinearity index,  $NLI$ . The metrics are defined as

$$NGI \triangleq \overline{\hat{b}^2} - \overline{\hat{b}_c^2}, \quad (3.82)$$

$$NLI \triangleq \left| \hat{b}_{\max}^2 - \left( \overline{\hat{b}^2} + 2\sigma_{\overline{\hat{b}^2}} \right) \right|, \quad (3.83)$$

with

$$\hat{b}^2(f_1, f_2) \triangleq \frac{|B(f_1, f_2)|^2}{E[|X(f_1)X(f_2)|^2] E[|X(f_1 + f_2)|^2]}, \quad (3.84)$$

$$B(f_1, f_2) \triangleq E[X(f_1)X(f_2)X^*(f_1 + f_2)], \quad (3.85)$$

where  $\hat{b}^2(f_1, f_2)$  is the squared bicoherence at frequencies  $f_1$  and  $f_2$ ,  $\overline{\hat{b}^2}$  is the aver-

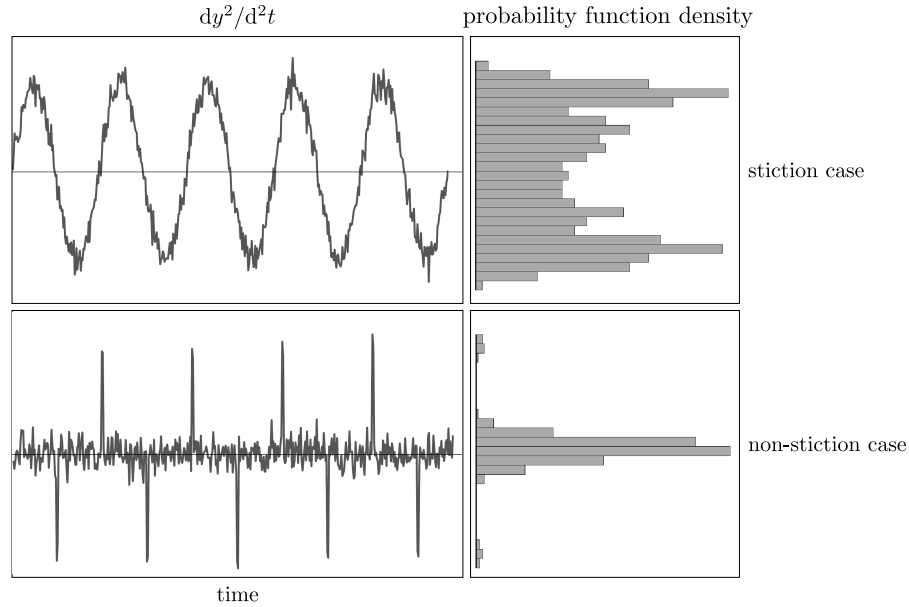


Figure 3.24: Probability density functions characteristic of the stiction and non-stiction cases (Horch, 2000).

age squared bicoherence,  $\overline{\hat{b}_c^2}$  is the statistical threshold (or critical) value obtained from the central chi-square distribution of the squared bicoherence,  $\hat{b}_{\max}^2$  is the maximum squared bicoherence,  $\sigma_{\overline{\hat{b}_c^2}}$  is the standard deviation of the squared bicoherence,  $X(f)$  is the discrete Fourier transform of the process output at frequency  $f$ ,  $B(f_1, f_2)$  is the bispectrum at frequencies  $f_1$  and  $f_2$ ,  $X^*(f)$  is the complex conjugate of the discrete Fourier transform  $X(f)$ , and  $E[\cdot]$  is the expectation.

This approach and those of Choudhury et al. (2006b,a) were later patented in Choudhury et al. (2007, 2012).

Singhal and Salsbury (2005) proposed a method to detect stiction in an oscillating control loop based on the calculation of the ratio between areas before and after the peak of an oscillating signal (Figure 3.26) using the quantity  $R$  defined as

$$R = \frac{A_1}{A_2} . \quad (3.86)$$

The decision rule is then summarized as: if  $R > 1$  the valve is sticking, but if  $R \approx 1$  the controller is aggressive.

The main principle is to recognize the shape produced by the control signal in a phase plot. The authors argued that the method is intuitive, requires very little

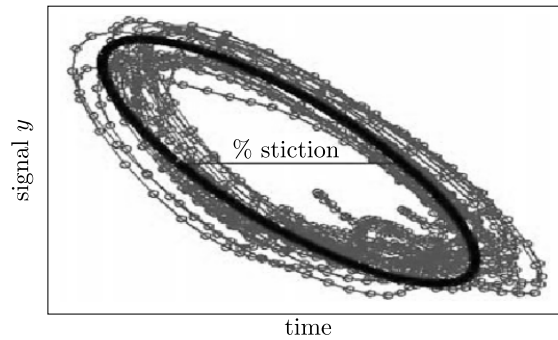


Figure 3.25: Detection of stiction through the fitting of an ellipse (Choudhury et al., 2006b,a).

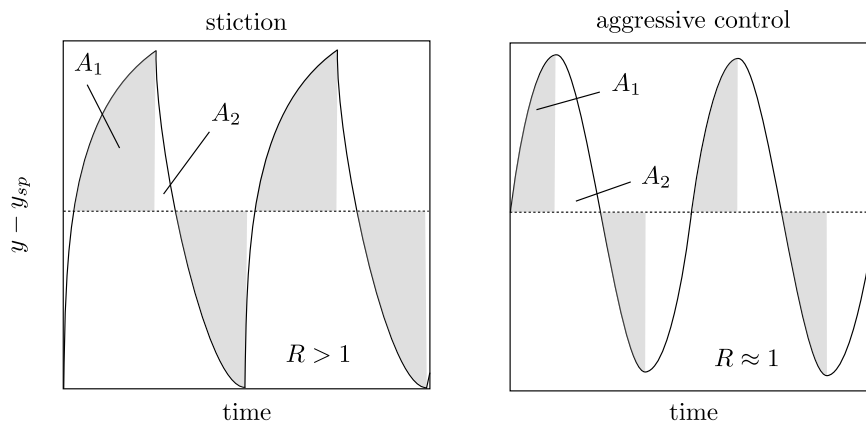


Figure 3.26: Shape of the control error for valve stiction and aggressive control. Adapted from Singhal and Salsbury (2005).

computational effort, and is easy to implement online in spite of some limitations that they also report: it is not applicable to integrating processes and does not distinguish stiction from other nonlinearities. As the method is based on calculating areas, some factors such as the signal noise and the sampling time need to be carefully considered.

Srinivasan et al. (2005a) suggested the comparison of the valve input versus controlled variable data shape with the sequences commonly seen in valves with stiction problems. The approach consisted of pattern recognition using the dynamic time warping technique to find the optimal alignment between two given sequences. The authors tested the method in different scenarios including non-constant behavior, intermittent stiction and external disturbances.

Zabiri and Ramasamy (2009) developed a method that calculates an index



based on nonlinear principal components analysis (NLPCA) using the distinctive shapes of the signals caused by stiction and other sources. The index is termed NLPCA curvature index and is calculated as

$$I_{\text{NC}} = \frac{\sum_{i=1}^n \overline{P_i P_{i+1}}}{\overline{P_1 P_n}}, \quad (3.87)$$

where  $P_i$  are the values of the loadings matrix obtained from NLPCA, and  $n$  is the number of observations.

Together with its coefficient of determination, the index quantifies the degree of nonlinearity and determines the presence of stiction. The method does not make assumptions on the control valve characteristics (e.g., a linear valve) since it does not assume any particular valve model. Although it is simple, effective and easy to implement, the authors observed ambiguity in the results for integrating processes as well as the need of a large amount of steady-state data for a correct detection.

Since the real valve position (valve output) is often not available in the data collected by the DCS (Distributed Control System), Chitralkha et al. (2010) developed an approach for estimating this variable through the application of the unknown input observer technique. After the estimation, they fit a trapezoid to the valve output versus valve input data, solving a constrained optimization problem to find the four corner points of the polygon. Although the method does not assume any specific stiction model, the authors used the Choudhury Model to prove the effectiveness of their method.

Also based on a shape analysis of the waves, Hägglund (2011) proposed a method that determines whether the shape of data between two consecutive zero crossings of the control error corresponds best to a sine or a square wave. To decide whether the loop has stiction, the author developed a normalized index that allows to infer the existence of the problem if a positive value is found. The index is given by

$$I_{\text{stiction}} = \frac{V_{\text{sine}} - V_{\text{square}}}{V_{\text{sine}} + V_{\text{square}}}, \quad (3.88)$$

where  $V_{\text{sine}}$  and  $V_{\text{square}}$  are loss functions in the following form

$$V_{\text{sine}} = \sum_{i=1}^n \left[ e(t_i) - a_{\text{sine}} \sin \left( \frac{2\pi}{T_P} i h \right) \right]^2, \quad (3.89)$$

$$V_{\text{square}} = \sum_{i=1}^n [e(t_i) - a_{\text{square}}]^2. \quad (3.90)$$

In (3.89) and (3.90),  $e(t_i) = y_{\text{sp}}(t_i) - y(t_i)$  represents the control error,  $h$  the sampling time,  $T_P = 2(t_{c1} - t_{c0})$  the sine wave period corresponding to the times of the zero crossing,  $t_{c0}$  and  $t_{c1}$  the two latest times of zero crossings of the control error,  $n = T_P/2h$  the number of samples in the interval  $[t_{c0}, t_{c1}]$ , and  $a_{\text{sine}}$  and  $a_{\text{square}}$  the amplitude of the pure sine and square waves. The procedure is automatic and may be performed off- or on-line.

Other methods are based on additional knowledge about the qualitative shape of the characteristic curve of the valve. For instance, Rengaswamy et al. (2001) developed a qualitative shape-based method in which the valve input data is fitted to find the most common types of oscillations: triangular, sinusoidal and square oscillations.

Also, Yamashita (2006a) proposed a method for the diagnosis of valve stiction based on the typical patterns of valve input versus valve output data. The valve movements are classified using the notation I (for increasing), D (for decreasing) and S (for steady). Some sequences of these letters represent the stiction pattern. The idea consists in counting the periods  $\tau$  of sticky movement and calculates the following indexes

$$\rho_1 = \frac{\tau_{\text{IS}} + \tau_{\text{DS}}}{\tau_{\text{total}} - \tau_{\text{SS}}}, \quad (3.91)$$

$$\rho_2 = \frac{\tau_{\text{IS II}} + \tau_{\text{IS SI}} + \tau_{\text{DS DD}} + \tau_{\text{DS SD}}}{\tau_{\text{total}} - \tau_{\text{SS}}}, \quad (3.92)$$

$$\rho_3 = \rho_1 - \frac{\tau_{\text{IS DD}} + \tau_{\text{IS DI}} + \tau_{\text{IS SD}} + \tau_{\text{IS ID}} + \tau_{\text{IS DS}} + \tau_{\text{DS DI}} + \tau_{\text{DS SI}} + \tau_{\text{DS ID}} + \tau_{\text{DS II}} + \tau_{\text{DS IS}}}{\tau_{\text{total}} - \tau_{\text{SS}}}. \quad (3.93)$$

Varying between 0 and 1, the indexes  $\rho_1$ ,  $\rho_2$  and  $\rho_3$  detect stiction if their values are greater than the threshold value 0.25.

Yamashita's method revealed excellent performance in detecting the stiction phenomenon in cases where other methods failed. Manum (2006) and Manum and Scali (2006) investigated the method performance using a large number of industrial flow control loops and concluded that the method correctly identifies the presence of stiction in 50% of the cases. These authors argue that one of the suggested indexes is not accurate enough to identify stiction. Another disadvantage of this method is that it requires valve stem data. Although this data is often unavailable, it is possible to apply the method for flow control loops where, assuming linearity and fast dynamics, the controlled variable is proportional to the real valve position.

Yamashita (2006b) addressed this disadvantage with the development of a new index for systems with slower dynamics, such as the level control loops. The index evaluates the excess kurtosis defined by

$$\gamma = \frac{1}{n} \sum_{i=1}^n \frac{(\Delta y_i - \mu_{\Delta y})^4}{\sigma_{\Delta y}^4} - 3, \quad (3.94)$$

where  $\Delta y$  is the differential of  $y$ ,  $\mu_{\Delta y}$  and  $\sigma_{\Delta y}$  are the mean and the standard deviation of  $\Delta y$ , and  $n$  is the number of observations of  $\Delta y$ . A loop suffering of stiction presents a two peaked distribution which means a negative large value of excess kurtosis.

Later, he extended the diagnostic method proposing a measure to quantify stiction in control valves (Yamashita, 2008). The degree of stiction is evaluated by calculating the width of the sticky pattern from the signals. This is an important development as it allows to prioritize the list of the control loops to be revised by the maintenance team.

Kalaivani et al. (2014) proposed a procedure to detect and quantify stiction using trends qualitative analysis based on ant colony optimization. Firstly, the method performs a piecewise fitting of the control signals. Triangular and sinusoidal waves are fitted to the controller output data and the parameters of the Stenman model are estimated using ant colony optimization by minimizing the error between the actual stiction model output and the simulated stiction model output.

### **System identification-based methods using the Hammerstein Model**

Recently, stiction detection / quantification developments were proposed by means of system identification using the Hammerstein Model. This is a commonly used model that is composed by a static nonlinear element in series with a linear dynamic part (Eskinat et al., 1991). The nonlinear element represents the sticky valve while the linear part models the process dynamics.

The first example is the approach of Stenman et al. (2003) based on the Stenman Model and on an ARX process model to detect stiction inspired by multi-model mode estimation techniques. In addition, the method does not require that an oscillating behavior of the loop be observed.

Srinivasan et al. (2005b) fitted the valve input and the controlled variable datasets to a Hammerstein Model defined also by the nonlinear Stenman Model plus a linear ARX model. As stiction nonlinearity is often modeled as a discontinuous phenomenon, the grid search algorithm was used to determine the only parameter of the chosen stiction model while the model parameters were computed through the separable least-squares method.

Lee et al. (2008) used the ordinary least-squares method to identify the whole Hammerstein Model. Other differences in their approach lie in the chosen stiction model (the He Model) and in the process structure assumptions (the first- or second-order plus time delay model). Additionally, their work defined a bounded search region for the stiction model parameters formulated as a constrained optimization problem. The low computational cost of the algorithm is one of the main advantages pointed out by its authors.

Choudhury et al. (2008b) improved the approach of Srinivasan et al. (2005b) by introducing the Choudhury Model, because the stiction model used by Srinivasan et al. (2005b) did not capture the true stiction behavior. The same two dimensional grid search method was used to estimate both the stiction and the process model parameters. Several variants of these approaches have been developed. Based on the opinion of Srinivasan et al. (2005b), according to which better search algorithms may be applied, Jelali (2008) developed a method using global optimization to estimate the parameters related to stiction phenomenon. The linear model parameters in this method were estimated using the least-squares identification technique. The method proved to be robust considering different

process types, controller settings and measurement noise. However, the high CPU time required in the estimation was a limitation.

Ivan and Lakshminarayanan (2009) introduced a modified identification approach based in the Hammerstein Model. The improvements over other previous works include the use of a modified He stiction model, a refined ARMAX model to identify the linear part, and the introduction of data pre-processing (such as data isolation and de-noising).

Karra and Karim (2009) considered a non-stationary disturbance term in the linear model through an extended ARMAX structure. This new term allows the inclusion of other possible root causes besides stiction, such as external disturbances. The work includes the nonlinear Kano Model for stiction description although alternative models are applicable, as well. Because of the discontinuity of the stiction phenomenon, the technique used in this work to determine all the Hammerstein Model parameters is the grid search algorithm.

Lee et al. (2010) developed a closed-loop method for stiction detection / quantification based on Hammerstein's modeling. It starts with the identification of the stiction model structure defining then the bounded search space of the stiction model parameters. Following this procedure, a constrained optimization problem is performed to identify the model parameters based on the mean-squared error criterion. The method was tested and validated with industrial data.

Following Lee et al. (2010) work, Qi and Huang (2011) built a bootstrap approach based on the Hammerstein Model identification to determine the confidence interval of the estimation. These researchers argued that the estimation of the stiction model parameters is not enough to quantify stiction and, consequently, may lead to incorrect conclusions. They proposed the calculation of the uncertainty of the estimated stiction parameters to complement the information achieved by model identification, making the diagnosis of the problem more reliable.

Another interesting work related to this subject is that of Srinivasan et al. (2012) who developed a reliability measure via frequency domain analysis of closed-loop systems to validate the results obtained from stiction detection methods based on the Hammerstein Model. This measure is calculated independently by the detection method and is applicable only for linear systems. It is note-

worthy that the methods based on the Hammerstein Model are not suitable to identify stiction unambiguously in integrating processes.

Babji et al. (2012) proposed a methodology where the Hammerstein Method and the Hilbert-Huang Transform are combined for root cause analysis. The Hammerstein Method developed by Srinivasan et al. (2005b) was used to detect and quantify stiction, while the nonparametric transform was used to distinguish oscillations occurring due to marginally stable control loop and external disturbances.

Shang et al. (2013) applied particle swarm optimization to estimate the parameters of the stiction model in a Hammerstein Model configuration where the nonlinear and linear blocks are described by the Chen Model and by an ARX model, respectively.

Brásio et al. (2014) proposed an approach for the detection and quantification of valve stiction using a one-stage optimization technique. A Hammerstein Model that comprises a complete stiction model (Chen Model) and a process model (first-order model) is identified from industrial process data. It is noteworthy that in order to simplify the identification process, the discontinuity of the stiction model is smoothed by a continuous function.

Lei et al. (2013) proposed the detection and quantification of stiction based on an extended Hammerstein Model where the discretized Preisach Model is used to capture the behavior of a sticky control valve and a linear dynamic model is used to describe the process. To identify the extended Hammerstein Model an iterative method is presented. In short, the method estimates the parameter vectors in two iterative steps in which the estimation problem is linear at each step. Also, Wang et al. (2014) adapted the identification algorithm to the new model and proved its identifiability, resulting in a more flexible structure to describe asymmetric stiction.

Also based on a Hammerstein Model (using the Kano Model for stiction modeling and an ARX model for process modeling), Bacci di Capaci and Scali (2014) presented a procedure that includes oscillation detection, stiction detection, data division and stiction quantification. Data division increases the reliability of the results by discarding data that may provide wrong quantification of the phenomenon. The method can be very useful in what concerns valve maintenance

scheduling and checking.

### **System identification-based methods using the Wiener Model**

While there are several studies discussing linear processes, nonlinear process control loops have not received the same attention despite all the potential benefits. One of the few approaches to tackle nonlinearity is based on the Wiener Model that is composed by a linear dynamic block connected to a nonlinear static part (Vörös, 2001).

Wang and Wang (2009) extended the study of Jelali (2010) using a generalized stiction-Wiener model to describe the valve stiction and the nonlinear process dynamics. These authors used the two-layer binary tree data-driven model proposed by Chen et al. (2008) and applied a novel global search grid identification algorithm for the quantification of stiction in closed-loops subject to colored noise. The feasibility of the algorithm is successfully illustrated by its authors in a case study.

Romano and Garcia (2010, 2011) associate the stiction phenomenon and the nonlinear process in parallel via the Wiener Model which deals with eventual external disturbances. The process model is decomposed in a linear and a nonlinear blocks. This structure received the designation of Hammerstein-Wiener Model. Although not explicitly mentioned in the model name, the stiction phenomenon is modeled using the Kano Model (model chosen by Garcia (2008) based on ISA standard tests). The linear block is represented by an ARMAX model while external disturbances are represented using transfer models of  $n$ th-order. For unknown nonlinear process dynamics, piecewise polynomials of third degree are used to model the nonlinear block. This approach utilizes the Nelder-Mead Simplex algorithm for searching the optimal pair of stiction model parameters leading to a reduction of the computational effort when compared with exhaustive searching algorithms. Even though the method has reasonable results, it should be simplified to make it suitable as a detection tool in industrial contexts. The large number of parameters to be estimated also may affect the method effectiveness when performed on-line.

### Other approaches

Ulaganathan and Rengaswamy (2008) also considered the nonlinearity of the process. The nonlinear Stenman Model is used in the first block to represent the stiction phenomenon. This block is connected to the process block characterized by a nonlinear dynamics. Finally, a linear external disturbance, modeled by a moving average model, is considered in the loop. The process is described by a second-order model.

Zabiri et al. (2009) adopted an algorithm incorporating a neural network to simultaneously identify the model parameters and quantify the stiction phenomenon. This approach, which uses the Choudhury Model for the stiction modeling, has the advantage of being applicable to all kind of processes. Also, Venceslau (2012) presented an artificial neural network approach in order to detect and quantify the amount of stiction using only the controlled variable and valve input information. The author applied different information preprocessing methods based on the calculation of centroid and Fourier Transform to facilitate the approach via neural networks. The main advantage of the method is the fact that it does not need any specific knowledge either of the process or of the valve output data.

Nallasivam et al. (2010) used the Volterra model-based technique to detect stiction in closed-loop nonlinear systems, the Stenman Model to represent the stiction phenomenon, and a known nonlinear process model to identify both the disturbance model parameters and the stiction model parameter. The grid search technique discussed by Srinivasan et al. (2005b) was successfully applied to obtain the stiction model parameter. These researchers estimated a moving average model for the disturbance. This strategy, applicable to nonlinear systems, has the peculiarity of not requiring prior information on whether the loop is linear or nonlinear.

Villez et al. (2010) proposed an active fault tolerant control strategy which enables the detection of valve stiction. The method, which showed promising results, is based on the Kalman Filter and assumes that the system is linear time invariant.

Farenzena and Trierweiler (2012) tackled the problem of stiction and backlash detection in integrating loops. They use the process variable patterns of a valve with backlash and stiction to detect the phenomena and even to distin-



guish between them. Computing first-order derivatives, they calculate an index for stiction diagnosis. Using several simulation cases, they proved the efficacy of the method through the correct detection of around 100% of the loops. Moreover, it is computationally inexpensive and requires only routine operating data. In spite of these considerable advantages of the method, its high dependency on the sampling time may lead to incorrect detection. Moreover, the tuning of the controller also affects the detection success.

The describing function method, a common tool to predict the period and amplitude of limit cycles in control loops, was applied by citearaujo-et al10 to detect and quantify a general nonlinearity using common process data and an approximate model of the nonlinearity. This approach showed good performance in the presence of model uncertainty and of multiple frequency oscillations. Later, the same authors applied the method to the specific nonlinearities such as deadband and stiction (Araujo et al., 2012), generalized the procedure for processes with unknown models, and provided the conditions for the uniqueness of the solutions as well as a sensitivity analysis to indicate situations where the error may increase. The method is a simple and efficient numerical algorithm that may be extended to other nonlinearities that cause limit cycles.

Other approaches applied the surrogate analysis to evaluate the nonlinearity of a signal. Thornhill (2005) developed a method to compare the signal and its surrogate data predictability using the index

$$N = \frac{\Gamma_{\text{surrogate}} - \Gamma_{\text{signal}}}{3\sigma_{\Gamma_{\text{surrogate}}}}, \quad (3.95)$$

where  $\Gamma$  is the mean square error, and  $\sigma$  is the standard deviation.

The surrogate data is useful because it provides a reference distribution against which the properties of the signal under test may be evaluated. Once the signal is more structured and more predictable than the surrogate data, the method evaluates the distribution properties of the original signal and of the respective surrogate data. The presence of a nonlinear signal may mean that stiction is a possible source of oscillation.

Alemohammad (2011) presented a stiction detection method designed for multi-loop control systems using both surrogate analysis and qualitative shape-based

approach. The method was applied on simulation and industrial data allowing to detect multiple sticky valves.

Based on the previously developed semi-physical model (He and Wang, 2010), He and Wang (2014) proposed a noninvasive valve stiction quantification method by using linear and nonlinear least-squares methods which are robust and easy to implement.

Most of the stiction detection methods assume that there exists oscillations in the control loops. This assumption may reduce the stiction detection robustness in the presence of system disturbances. Zakharov and Jämsä-Jounela (2014) proposed an oscillation detection method that evaluates the similarity of the oscillation periods by means of a correlation coefficient and compared it against five other methods reported in the literature. The work also introduced two indexes to quantify the mean-nonstationarity and the presence of noise in oscillating signals.

Arumugam (2014) presented an adaptive neuro-fuzzy methodology for the identification of stiction in a vertical two tank process based on the Kano Model.

### ■ Stiction compensation

The most effective solution for a sticky valve is to repair it. However, this may not be feasible between plant turnarounds and, therefore, alternatives to mitigate the negative impact on the plant should be considered. These alternatives consist in trying to compensate the stiction phenomenon. Several compensation algorithms have already been developed in process control and also in other areas such as robotics (Bona and Indri, 2005).

The first classification of stiction compensators by Dupont et al. (2002) divided them into model and non-model based. Recently, Sivagamasundari and Sivakumar (2013) proposed a more detailed classification that is further updated herein (see Figure 3.27) by adding a new branch that contains approaches that were left out in the previous classifications.

Although the non-model based compensators do not directly use a model, they require one for the prediction of operating point stability, of limit cycle stability or for performance analysis (Dupont et al., 2000). This promoted the development of feedforward and feedback strategies relying on stiction models to

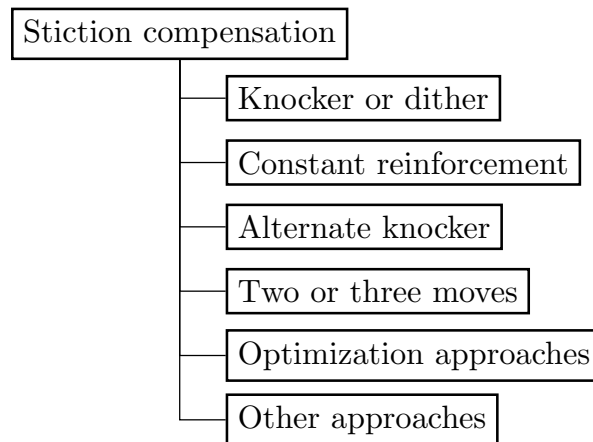


Figure 3.27: Compilation of the compensation methods.

cancel the stiction force (Geffen, 2009). Canudas de Wit et al. (1995) rightfully assert that the success of these compensators depends on the appropriateness of the model structure and on the knowledge of the model parameters. Also, the capability for parameter estimation is important when dealing with adaptive controllers. These authors created a stiction compensation strategy that adds a signal calculated with an observer built using the LuGre Model.

Canudas de Wit et al. (1987), Friedland and Park (1992), Ro and Hubbel (1993), Amin et al. (1997) and Feemster et al. (1998) used adaptive control algorithms that are based on static first-principle models to compensate stiction. A stiction compensating adaptive controller based on the Dahl Model was designed and used by Walrath (1984) and Leonard and Krishnaprasad (1992). Adaptive compensation methods using the LuGre Model were also proposed by Canudas de Wit and Lischinsky (1997), Panteley et al. (1998) and Altpeter et al. (2000). Canudas de Wit (1993) suggested a different control scheme aiming to strengthen the closed-loop system in the cases where friction overcompensation exists. It takes into account the uncertainties in the parameters and in the structure of the friction model that may lead to inexact friction compensation in servo-mechanisms. This control scheme reduced the oscillations in amplitude and modified the frequency independently of the closed-loop system specifications.

Motivated by the complexity inherent to stiction models and the ability of neural networks to approximate nonlinear behavior, Otten et al. (1997) designed a neural network based controller for stiction compensation considering a static

model.

Panteley et al. (1998) developed a novel adaptive friction compensator based on the dynamic Dahl Model ensuring global position tracking when applied to a system subject to friction forces. This method considers that all the parameters (of the system and friction models) are unknown and uses a very simple adaptive law to estimate them requiring only measurements of valve position and velocity.

As a generalization of the method suggested by Vedagarbha et al. (1997), Hirschorn and Miller (1999) developed a Lyapunov based continuous dynamic controller as a substitution of a tuned PID controller for nonlinear systems using the Dahl Model. This method removes the restrictions on pole locations of the compensated system allowing to improve the velocity of its response. However, the authors observed that the overestimation of some parameters caused degradation in performance.

Kayihan and Doyle-III (2000) also addressed the subject developing an algorithm for stiction compensation that reconstructs the unmeasurable states providing a robust control action. The algorithm uses the Classical Model to describe stiction and assumes that all model parameters are known. Unfortunately, such detailed valve information is often not available.

Huang et al. (2000b) compensated the effects of stiction using an adaptive scheme coupled with the LuGre Model which also incorporates a neural network to model stiction. The technique utilizes a PD control structure and an adaptive estimation of the friction force to correct the effects prediction. The results were highlighted with a simulation experiment.

Lampaert et al. (2004) combined a model-based compensation method and a disturbance observer to develop a new approach to compensate stiction. Static friction was modeled using the GMS Model. Because both approaches are complementary, the combination resulted in an accurate tracking performance.

In general, these compensation methods utilize complex models to describe the stiction phenomenon and that restricts their industrial use. Nevertheless, the most recent methods tend to be simpler. They may be classified into the categories below.

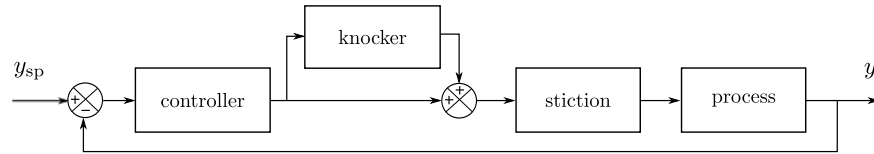


Figure 3.28: Diagram illustrating the knocker used in a feedback loop (Hägglund, 1997; Hägglund, 2002).

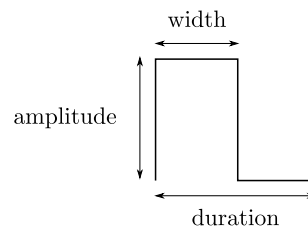


Figure 3.29: Signal sent by the knocker compensator (Hägglund, 1997; Hägglund, 2002).

### Knocker or dither

The knocker approach, also known as dither approach, consists of adding a high frequency signal to the control signal before it is input to the system under a feedback loop with the purpose of preventing process output fluctuations (Figure 3.28).

Hägglund (1997); Hägglund (2002) developed the first knocker compensation method specifically targeting stiction in control valves. This researcher added a pre-designed signal to the valve input so that the oscillations produced by stiction are minimized. The signal shown in Figure 3.29 consists of short pulses with constant amplitude, width and duration that must be tuned. His method successfully removed oscillations induced by stiction from the controlled variable at the cost of a faster and wider motion of the valve stem, i.e., increasing the rate of mechanical wear of the valve. In fact, this aggressive movement and the required parameters tuning are the main drawbacks of the method. To overcome such disadvantage, Srinivasan and Rengaswamy (2005, 2006b) developed a framework implementing some suggestions for the automated choice of the parameters. The framework, that integrates detection and compensation tasks, showed to reduce the output variability found in the studied systems by 6 to 7 times.

de Souza L. Cuadros et al. (2012b) proposed a method for stiction compensation that is also based on the knocker approach. The authors added a supervision layer to analyze the control error and interact with the PID controller. They verified that its performance is good in simulation and experimental tests. The strategy showed a reduced integral absolute error and a lower number of valve movements.

All these knocker methods are characterized by a faster motion of the valve and, thus, may cause mechanical problems. Therefore, they are just short-term solutions.

### **Constant reinforcement**

Ivan and Lakshminarayanan (2009) suggested an alternative similar to the knocker approach. They preferred to design the compensating signal as a constant and to add the reinforcement to the valve input only when this variable is not constant. The constant compensation value recommended by the authors is the stiction parameter estimated by the detection method suggested in the same work. This value is changed by the sign of the valve input variation. Although the method is very useful for reduction of the output variability associated to a sticky valve, it does not decrease the valve aggressiveness.

### **Alternate knocker method**

Srinivasan and Rengaswamy (2007) proposed the addition of a special block to the nominal PID algorithm. However, this control signal adaptation is not known by the nominal controller and, consequently, it affects negatively the performance of the controller. Moreover, since it is not taken into consideration at the time of controller commissioning, the tuning parameters that are determined without stiction compensation may even produce instability and/or additional wear of the valve and the actuator.

### **Two or three moves compensator**

The main focus of the two moves compensation method first introduced by Srinivasan and Rengaswamy (2008) is to maintain the valve at its steady-state position. To achieve this objective, at least two stem moves in opposite directions

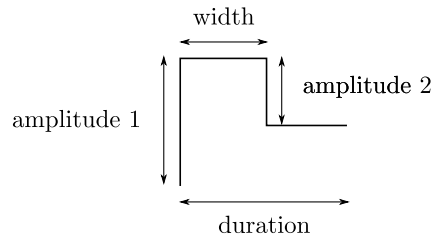


Figure 3.30: Signal sent by the two moves compensator (Karthiga and Kalaivani, 2012).

are required as is shown in Figure 3.30. The compensating signals should have magnitudes large enough to overcome stiction and move the valve stem, but not sufficient to saturate it. This method has some limitations related with the change of the control loop setpoint preventing its implementation on an automated setpoint tracking. Also, it uses a simpler model (the Stenman Model) to describe the stiction phenomenon, decreasing its accuracy.

More recently, Farenzena and Trierweiler (2010) proposed a novel methodology to compensate stiction effects, through the modification of the PI controller block in the control loop. Instead of adding a compensator block, the approach adapted the existent and traditional PI controller block regarding the stiction phenomenon. Similarly to Srinivasan and Rengaswamy (2008), the authors used a two move method that allows to specify closed-loop performances faster than open-loop and reject load disturbances efficiently. In spite of this similarity, the method is able to track the setpoint, unlike the method of Srinivasan and Rengaswamy (2008). Beyond that, the method admits the existence of a small offset from the setpoint, reducing significantly the valve travel.

Srinivasan and Rengaswamy (2008) proposal was thought for first-order non-integrating systems. Later, Lanfredi (2011) applied the method also to second-order integrating systems with good results in the reduction of the control error and of the variability of the stem valve.

de Souza L. Cuadros et al. (2012a) revised the Srinivasan and Rengaswamy (2008) method and suggested two improved versions to circumvent the drawback related to the setpoint tracking. None of the methods require the knowledge of the plant model and both may handle setpoint changes by detecting the increase of the control error. In spite of these advantages, the first method is still

susceptible to disturbances. The second method is more robust since it was especially developed to overcome this aspect. The methods drawbacks are related to the requirement of having similar control valve and process dynamics.

A new compensation method based on two movements of the valve was also developed by Wang (2013). The method adds a short-time rectangular wave to the setpoint in two distinct movements. The method is applied to a control loop operating in automatic and leads the valve to the desired position avoiding high variability. The main advantage of this method is its robustness against modeling errors and measurement noise.

Besides the comparison between several compensation methods, Silva and Garcia (2014) developed an improved two moves approach merging the two moves and the constant reinforcement methods. In spite of providing similar results to those obtained by other compensation methods, the developed algorithm is harder to implement and has the necessity to disable the PID controller for short periods of time which is not practical.

Karthiga and Kalaivani (2012) proposed a similar stiction compensation method involving not two but three movements that revealed to be capable of reducing the valve travelling. This method exhibits a lower overshoot and settling time than other methods. The authors showed, by simulation, that the method imposes a smoother valve operation, resulting in a longer control valve life.

### **Optimization approaches**

Srinivasan and Rengaswamy (2006a, 2008) proposed an optimization based approach for stiction compensation attempting to meet a less aggressive stem movement, reduced output variability and less energy in the signal added to the control signal. A cost function is built taking these aspects into account and its value is minimized using the compensator moves as optimization variables. This technique was used with the Stenman Model. The authors observed significant improvements compared to the classical approaches but stressed the need for analyzing the model mismatch effect, the incorrect stiction measurement, and the real time issues before trying an online implementation. In addition, the method is computationally more expensive when applied to a large number of sticky valves. As the cost function is non-smooth, the optimizer may be not able to



attain a global minimum and the process output may not reach the setpoint.

Based on Hägglund (2002) and Srinivasan and Rengaswamy (2005, 2006b), Sivagamasundari and Sivakumar (2013) proposed a new compensation approach for the stiction nonlinearity present in control valves. The model-based approach uses the He Model to determine the stiction magnitude since it predicts the behavior of the problematic valve more precisely. Instead of tuning the parameters that determine the wave form of the compensated signal, the authors developed a few rules to find those parameters. The method tuned with these rules achieved a non-oscillatory output without forcing faster and wider moves of the valve stem. In addition, it does not need process or controller extensive information and allows a good tracking of the setpoint changes during operation.

### **Other approaches**

Gerry and Ruel (2001) suggested simple and practical techniques for tackling stiction on-line. Basically, they proposed a set of tuning rules to reduce the effect of the stiction-induced oscillations at the cost of steady-state control errors.

Márton and Lantos (2007) reformulated the control law adding a term that guarantees good tracking while cancels the effect of the friction force. Using the model developed in the same work, the stiction model parameters are re-tuned adaptively to account the modeling errors.

Halimi and Kune (2010) introduced a block in the control loop consisting in a filtered feedback signal added to the control signal. They claimed that this method may reduce significantly the oscillations in the controller variable (up to 75%). However, the authors did not specify either the structure or the order of the filter introduced in the loop.

Alemohammad and Huang (2012) proposed a compensation framework based on the oscillation condition previously introduced (Alemohammad and Huang, 2011). This condition allows to predict the occurrence and the severity of stiction-induced oscillations in control systems, based on which it is possible to reduce or eliminate process oscillations by following some controllers re-tuning guidelines suggested by the authors. Fang et al. (2015) also developed an approach based on the PID controller tuning. It reduces the oscillation amplitude generated by the stiction phenomenon tuning the controller parameters via a complex equa-

tion that has as inputs the He Model parameters, the process model (first-order plus time delay) parameters and two other variables calculated by an iterative method.

Finally, Zabiri and Samyudia (2009) proposed a model predictive control formulation that is based on mixed integer quadratic programming and showed that the closed-loop performance may be significantly improved if stiction is taken into account explicitly in the optimization problem. However, the approach requires that the stiction parameters be known a priori. Besides, the MIQP formulation may not work well in highly nonlinear or highly dimensional systems because of the required computational burden and the resulting feedback latency.

Silva and Garcia (2014) developed an experimental comparison study of stiction compensation methods using such metrics to evaluate their performance as the integral absolute error, a factor related to the stem position variation, a factor related to the valve actuator pressure variation, and the rising time. The methods tested in the flow control loop of a pilot plant were applied to setpoint tracking and regulatory experiments. Although several methods exhibited good compensation capacity, the choice of the best method depends on the trade-off between the setpoint tracking, the stem position and actuator pressure variabilities, and the rising time.

### ■ Academic and commercial software

Most of the software packages that include features for stiction analysis are related to the detailed assessment of the control loop performance. Such software is specially important given the large number of control loops existing in industrial processes. Besides, these tools should not only determine the control loops that need maintenance but also propose the most appropriate remedy. Some of the most recent software packages for the assessment of the control loops performance are summarized in Table 3.4.

From the review presented so far, it is clear that there exists a wide variety of different techniques to analyze stiction in control valves. Even though most packages in Table 3.4 include stiction analysis, little has been done on integrating the different techniques.

Table 3.4: Performance assessment software (extended and updated from Shardt et al. (2012)).

Software	Company	Features
<i>Process Assessment Technologies and Solutions</i> <sup>1</sup>	University of Alberta, Canada	Performance assessment (univariate and multivariate analysis, minimum variance index and LQG benchmark) Economic performance assessment Data-driven performance assessment Stiction modeling, detection, and compensation
<i>Control Loop Performance Assessment</i> <sup>2</sup>	Petroleum University of Technology, Iran	Performance assessment (statistical measures) Valve analysis
<i>Control Performance Monitor (ProcessDoctor)</i> <sup>3</sup>	Matrikon/Honeywell	Performance assessment (control charts) Oscillation analysis Stiction detection and quantification Economics analysis Closed-loop system identification
<i>PlantTriage</i> <sup>4</sup>	Expertune	Oscillation analysis Performance assessment (minimum variance and settling time method) Stiction detection Root cause analysis Economic prioritisation of loop issues PID controllers tuning
<i>PCT Loop Optimiser Suite</i> <sup>5</sup>	Leikon GmbH	Performance assessment
<i>LPM, Loop Performance Manager</i> <sup>6,7</sup>	ABB	Auditing of loops Stiction detection Performance assessment (minimum variance)
<i>LoopScout</i> <sup>8,9</sup>	Honeywell	Stiction detection Performance assessment PID controllers tuning Oscillation detection
<i>EnTech Toolkit (DeltaV Inspect)</i> <sup>10</sup>	Emerson Process Management	Performance assessment (basic indexes)
<i>INTUNE</i> <sup>11</sup>	ControlSoft	PID controllers tuning Performance assessment
<i>Aspen Watch Performance Monitor</i> <sup>12</sup>	AspenTech	Performance assessment (statistical methods) PID controllers tuning Stiction detection
<i>Control Monitor</i> <sup>13</sup>	Control Arts, Inc.	Performance assessment (statistical and model based methods) Oscillation detection Stiction detection
<i>Control Loop Optimisation</i> <sup>14</sup>	PAS	Performance assessment Interaction detection Oscillation detection
<i>Condition Data Point Monitoring</i> <sup>15</sup>	Flowserve	Stiction detection
<i>Automatic Control Loop Monitoring and Diagnostics</i> <sup>16</sup>	PAPRICAN	Performance assessment Oscillation detection
<i>Plantstreamer Portal</i> <sup>17</sup>	Ciengis	Performance assessment Stiction detection and quantification

<sup>1</sup> Shardt (2014), <sup>2</sup> Salahshoor et al. (2011), <sup>3</sup> Matrikon (2012), <sup>4</sup> ExperTune (2012), <sup>5</sup> GmbH (2012), <sup>6</sup> Belli et al. (2006), <sup>7</sup> Horch et al. (2007), <sup>8</sup> Desborough and Miller (2002), <sup>9</sup> Honeywell (2012), <sup>10</sup> Emerson (2012), <sup>11</sup> ControlSoft (2012), <sup>12</sup> Aspen Tech (2012), <sup>13</sup> Control Arts (2012), <sup>14</sup> PAS (2012), <sup>15</sup> Flowserve (2012), <sup>16</sup> Paprican (2012), <sup>17</sup> Ciengis (2015).

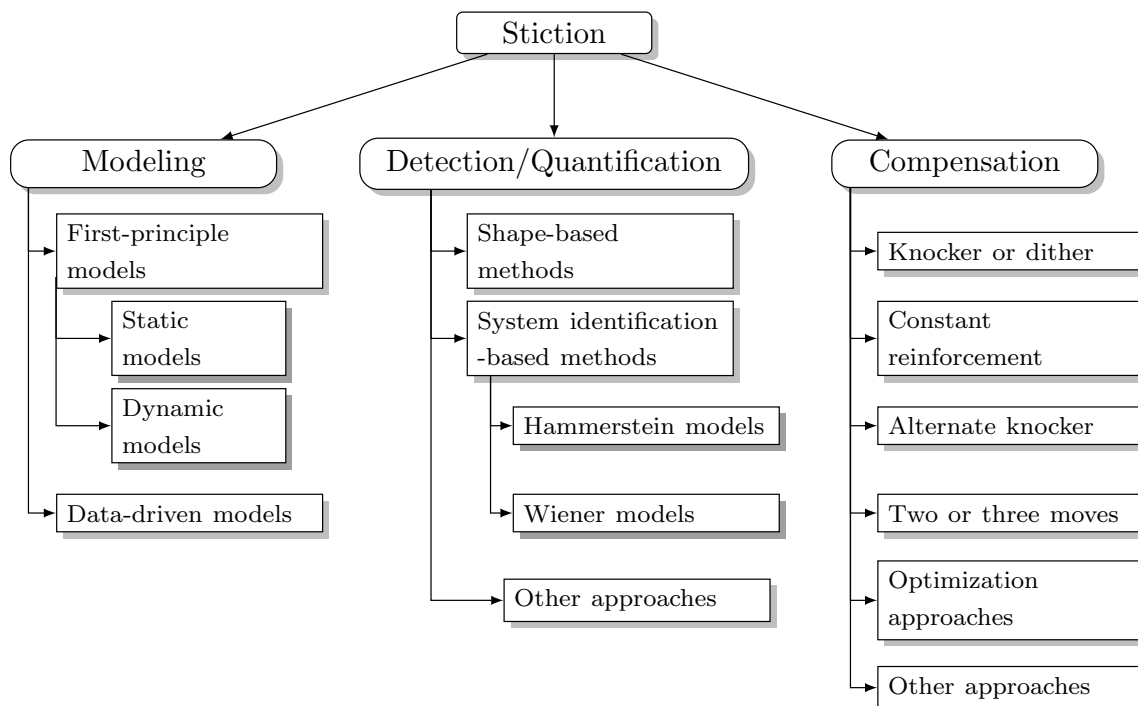


Figure 3.31: Family tree of models and detection / quantification and compensation methods of stiction.

### ■ Summary

Control loops are important capital assets and should be monitored to maintain the process safe and efficient. Control valves are key components of control loops and their malfunctioning deteriorates their performance. Stiction, as one of the most long-standing problems in process industry, has received a special attention in both industrial and academic environments. The present article reviews and systematizes the extensive work that has been developed by different authors in modeling, detection / quantification, and compensation of stiction. The result of such systematization is summarized in Figure 3.31.

The reviewed stiction models fall into two main classes: first-principle and data-driven. The former provide a detailed description of the most significant effects occurring in the valve. But the existence of a large number of often unknown parameters limits their use and, consequently, their application in stiction detection. The data-driven models reduce the complexity of the first-principle based modeling by capturing only the relevant behaviour observed in valves affected by stiction. These models are quite common in practice owing to the simplicity

and ease of use. However, their accuracy deteriorates when data extrapolation is attempted.

Regarding detection and quantification approaches, the published methods perform either a qualitative or a quantitative evaluation of stiction presence and magnitude. Most of the algorithms use available routine data from the control system. A number of methods require the knowledge of the real valve stem position and that severely limits their applicability. The quantification is often performed simultaneously with detection, especially in the case of model-based approaches.

Stiction compensation techniques are normally applied to mitigate the performance degradation of the control loop until valve repair is possible. Unlike the detection /quantification methods, the compensation methods are few and limited. The most studied and industrially applied compensation technique is the knocker that introduces an additional signal in the valve input variable providing the necessary driving force to overcome static friction. Interesting new methods based on optimization and on the changing of the control law are quite promising because they use quantitative metrics and adaptivity to minimize the impact of stiction. However, the difficulty in industrial application of these methods lies in the fact that they require global and/or mixed integer nonlinear optimizers and call for changes of the standard control algorithms implemented in control systems.

### **3.3 Detection and characterization of oscillating disturbances**

Oscillations induce an undesirable increase of the process variability leading to excessive production costs and to the decrease of the final product quality. Besides, their presence compromises the production stability and safety. It is therefore desirable to perform continuous monitoring in order to detect in real time control loops with oscillations (Brásio et al., 2014).

Since oscillations occur frequently in control loops and have harmful consequences, automatic detection has received some attention of the industrial and scientific communities.

The existing algorithms can be classified according to the number of time series (i.e., industrial datasets) needed for by the method. The methods aggregating multiple time series (Thornhill et al., 2002; Xia and Howell, 2005; Xia et al., 2005, 2007; Tangirala et al., 2007; Zang and Howell, 2007; Tangirala et al., 2005; Jiang et al., 2006b) do not detect the significant oscillation periods in the data. These methods identify the time series containing similar time patterns and determine the control loop that generates and propagates those patterns. Consequently, a first individual identification of the control loops affected by oscillations is crucial before applying the methods that determine the source of those oscillations.

Several developments have been made in what concerns the individual study of time series. Nevertheless, those methods present characteristics that lead to an incomplete detection and/or characterization of oscillations in control loops. The majority of them is unable to deal with slower trends, noise presence, and signals containing multiple oscillations which often characterize in industrial signals. Therefore, those methods are very dependent on their parameters tuning demanding a careful manual determination. In the present work, a new approach to analyse the industrial control loops performance by the detection and characterization of oscillations is developed. Using data easily available in plants, the new method detects multiple oscillations, is computationally light, and runs automatically.

### 3.3.1 Auto-correlation function

Consider the following definitions.

**Definition 1** The time series  $x_t$  is (weakly) stationary if (Brockwell and Davis, 2002, Definition 1.4.2)

- i. the mean  $\bar{x}$  is independent of  $t$ ,
- ii. the auto-covariance function  $\gamma_x(t, t + \tau)$  is independent of  $t$  for each  $\tau$ .

■

**Definition 2** Consider the stationary time series  $x_t$ . The auto-covariance function of  $x_t$  for the time delay  $\tau$  is given by

$$\gamma_x(t, t + \tau) = E[(x_t - \bar{x})(x_{t+\tau} - \bar{x})] , \quad (3.96)$$

where  $E[z]$  is the expected value of the generic variable  $z$ . In addition, the auto-correlation function of  $x_t$  is defined by

$$\rho_x(t, t + \tau) = \frac{\gamma_x(t, t + \tau)}{\gamma_x(t, t)} , \quad (3.97)$$

for a given value of  $\tau$  (Brockwell and Davis, 2002, Definition 1.4.3).

■

Equation (3.97) may be extended in the following way

$$\begin{aligned} \rho_x(t, t + \tau) &= \frac{E[(x_t - \bar{x})(x_{t+\tau} - \bar{x})]}{E[(x_t - \bar{x})(x_t - \bar{x})]} \\ &= \frac{E[(x_t - \bar{x})(x_{t+\tau} - \bar{x})]}{E[(x_t - \bar{x})^2]} \\ &= \frac{\frac{1}{N-\tau} \sum_{t=1}^{N-\tau} (x_t - \bar{x})(x_{t+\tau} - \bar{x})}{\frac{1}{N} \sum_{t=1}^N (x_t - \bar{x})^2} . \end{aligned} \quad (3.98)$$

Some authors assume that the numerator of (3.98) may be approximated by

$$\frac{1}{N} \sum_{t=1}^{N-\tau} (x_t - \bar{x})(x_{t+\tau} - \bar{x}) ,$$

a form with more advantageous statistical properties (NIST and SEMATECH, 2012). The direct substitution of this approximation into (3.98) originates the definition of the auto-correlation function used in the present work:

$$\rho_x(t, t + \tau) = \frac{\sum_{t=1}^{N-\tau} (x_t - \bar{x})(x_{t+\tau} - \bar{x})}{\sum_{t=1}^N (x_t - \bar{x})^2} , \quad (3.99)$$

where  $\bar{x}$  is the mean of the sample with size  $N$ ,  $\tau$  is the number of intervals corresponding to the time delay of the series, and  $\rho_x \in [-1, 1]$ .

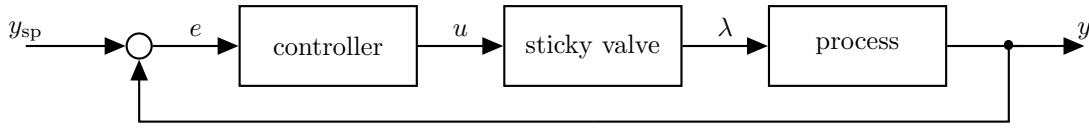


Figure 3.32: Process block diagram.

Gaussian noise  $w_t$  may be defined as a random time series with zero mean and variance  $\sigma_w^2$  being usually represented by the normal distribution  $w_t \sim \mathcal{N}(0, \sigma_w^2)$ . Given its properties, Gaussian noise is a stationary signal and its auto-covariance function may be defined as

$$\gamma_w(t, t + \tau) = \begin{cases} \sigma_w^2, & \text{if } \tau = 0 \\ 0, & \text{if } \tau \neq 0 \end{cases}, \quad (3.100)$$

and does not depend on  $t$  (Brockwell and Davis, 2002, Examples 1.4.1 and 1.4.2). By Definition 2, its auto-correlation function is calculated as

$$\begin{aligned} \rho_w(t, t + \tau) &= \frac{\gamma_w(t, t + \tau)}{\gamma_w(t, t)} \\ &= \frac{0}{\sigma_w^2} = 0, \end{aligned} \quad (3.101)$$

that is, the Gaussian noise auto-correlation function is null for  $\tau > 0$ .

### 3.3.2 Proposed approach

The new method proposed herein automatically detects and characterizes the oscillations present in time series from that control loops.

A typical industrial control loop is composed by a Proportional-Integral-Derivative (PID) controller, a final control element (usually a control valve), and a process (which one intends to control), as represented in Figure 3.32. All the controller order to the final control element (manipulated variable or  $u$ ), the real position of the final control element ( $\lambda$ ) and the controlled variable ( $y$ ) may contain oscillations.

The main idea of the method consists in the detection of the different oscillations present in  $u$ ,  $\lambda$ , and  $y$  time series as well as their characterization by deter-



mining their oscillation periods. With this purpose, it applies the auto-correlation function to the time series. After the detection and characterization of the first period, the method automatically generates a new signal reducing the number of points of the time series and repeats the analysis searching for a new oscillation of higher period.

Sets of  $N$  points respecting to variables  $y_{sp}$ ,  $u$ ,  $\lambda$ , and  $y$  of the previously selected industrial control loops (among the active control loops of the plant) are read with a sampling time of  $\Delta t$  and stored chronologically in order to build the respective time series. Consider that each of these time series may be represented by a generic stationary time series of finite size  $N$  in discrete time points  $t$  constituted by  $x_t = \{x_1, x_2, \dots, x_N\}$  contaminated by the Gaussian noise  $w_t \sim N(0, \sigma_w^2)$ . The auto-correlation function of  $x_t$  is a measure of the statistical dependence between the series values at different times and is defined mathematically by (3.99).

The auto-correlation function of an oscillatory time series  $x_t$  corrupted by Gaussian noise  $w_t$  generates a signal that is also oscillatory and has the same period. Since the auto-correlation function of the noise  $\rho_w$  is null, the utilization of this function advantageously decreases the detrimental impact of the noise allowing for a more exact determination of the periods of the oscillations present in the time series.

Based on these considerations, the method depicted schematically in Figure 3.33 is proposed. The method consists of the following steps:

- Step 1. Determination of the auto-correlation function  $\rho_x$  via (3.99) with  $\tau = 1, \dots, N - 1$ .
- Step 2. Verification of the existence of oscillations using the stopping criterion SC1.
- Step 3. Determination of the smallest period of the oscillations present in the auto-correlation function.
  - (a) Determination of the time values  $t_i$  ( $i = 1, \dots, M$ ) for which the auto-correlation function is maximum.
  - (b) Verification of the existence of sufficient information using the stopping criterion SC2.
  - (c) Calculation of the oscillation periods  $T_i$  by

$$n_i = t_{i+1} - t_i, \quad (3.102)$$

$$T_i = n_i \Delta t, \quad (3.103)$$

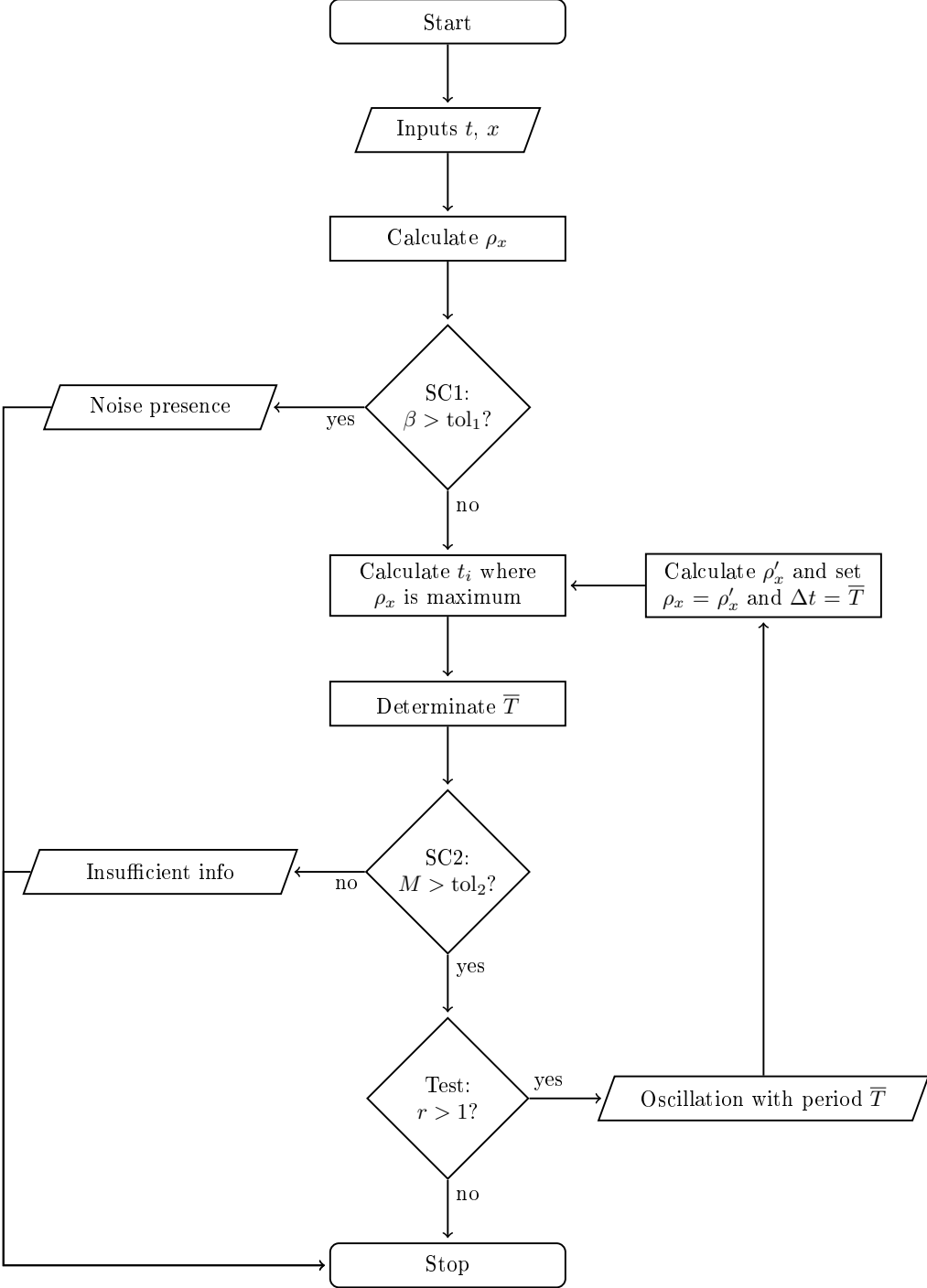


Figure 3.33: Proposed method flowchart.

where  $i = 1, \dots, M - 1$  and  $n_i$  is the number of time series points corresponding to a single oscillation.

(d) Calculation of the statistical parameters

$$\bar{n} = \text{int}\left(\frac{1}{M-1} \sum_{i=1}^{M-1} n_i\right), \quad (3.104)$$

$$\bar{T} = \frac{1}{M-1} \sum_{i=1}^{M-1} T_i, \quad (3.105)$$

$$\sigma_T = \sqrt{\frac{1}{M-2} \sum_{i=1}^{M-1} (T_i - \bar{T})^2}, \quad (3.106)$$

where  $\bar{n}$  is the mean number of time series points corresponding to a single oscillation,  $\bar{T}$  and  $\sigma_T$  are the mean and the standard deviation of the oscillation periods, respectively.

(e) Calculation of the statistical regularity  $r$  by

$$r = \frac{\bar{T}}{3\sigma_T}. \quad (3.107)$$

Step 4. Detection of oscillations by applying the test

$$\begin{cases} \text{If } r > 1, & \text{detected oscillation of period } \bar{T}. \\ \text{If } r \leq 1, & \text{no oscillation is detected.} \end{cases}$$

Step 5. Repeat of steps 3 and 4 using the auto-correlation function  $\rho'_x$  obtained by removal of the already detected oscillation period  $\bar{T}$  by

$$\rho'_{x,i} = \frac{1}{\bar{n}} \sum_{k=1+\bar{n}(i-1)}^{\bar{n}} \rho_{x,k}, \quad (3.108)$$

where  $i = 1, \dots, \#\rho_x/\bar{n}$  with  $\#\rho_x$  representing the number of points of  $\rho_x$ . The sampling period is set to  $\Delta t = \bar{T}$ .

The two stopping criteria referred as SC1 and SC2 are defined according to:

SC1. Oscillations presence: When data is not correlated, the auto-correlation function is null indicating the exclusive existence of Gaussian noise in the signal. In the proposed method, the detection of these situations in time series makes use of the confidence interval concept. Accordingly, if a large part of the data points of the auto-correlation function coincides in the confidence

interval, it is concluded that the points do not present temporal dependence and therefore the signal does not contain oscillations. The criterion is mathematically expressed by

$$\beta = \frac{\#\{i < N; |\rho_{x,i}| < \rho_{x,\text{lim}}\}}{N - 1} > \text{tol}_1 \quad , \quad (3.109)$$

with

$$\rho_{x,\text{lim}} = \frac{Z_{1-\alpha/2}}{\sqrt{N - 1}} \quad , \quad (3.110)$$

where  $\beta$  represents the percentage of points of  $\rho_x$  that are coincident in the confidence interval and  $\#S$  defines the number of elements in set  $S$ . The confidence interval limits for the auto-correlation function are calculated by  $\pm\rho_{x,\text{lim}}$ , where  $Z_{1-\alpha/2}$  is the cumulative distribution function for the standard normal distribution. For a confidence interval at 95% ( $\alpha = 0.05$ ),  $Z_{1-\alpha/2} = 1.96$ .

SC2. Sufficient information: In order to determine the oscillation period, at least four intervals are necessary (an interval corresponds to the distance between two consecutive zero crossings of the auto-correlation function) because otherwise the calculation of the period and of the standard deviation becomes unreliable. This criterion is defined by

$$M > \text{tol}_2 \quad , \quad (3.111)$$

where  $M$  is the number of intervals.

Using data easily available in plants, the method should be applied individually to each time series calculating the auto-correlation function of that series only once and identifying its maxima to detect and characterize the multiple oscillation possibly present in the signal. The method automation is of special importance and is accomplished using stopping criteria that allow to: (i) detect, still at an initial phase, the absence of oscillations and (ii) search for other oscillations while data quality is still sufficient.

Table 3.5: Methods analysed.

Method	Designation
Hägglund (1995)	A
Forsman and Stattin (1999)	B
Miao and Seborg (1999)	C
Wang et al. (2013)	D
method here suggested	E

### 3.3.3 Discussion of results

In this section, the performance of the method just described (Section 3.3.2) is studied. Moreover, a comparative analysis of different methods (including the new approach) is performed, emphasizing the advantages and disadvantages of each of them. Furthermore, the comparative study of the methods comprises not only applications to simulated data but also to data collected in industrial environments.

With such purpose, several methods available in the literature as well as the newly developed method were coded in the programming language GNU Octave. The five methods that were implemented are listed in Table 3.5. For the sake of compactness and clearness of the results presentation and discussion, an uppercase letter was assigned to each of the methods under analysis to work as their short designations.

Each method is tested using data from two different origins:

**Simulated data:** Three different scenarios common in process industry (see Figure 3.34) were considered to generate data: a healthy control loop, D1; a loop affected by an oscillation (of period 20 s), D2; and a loop suffering of the overlay of two oscillations (of periods 20 and 200 s), D3. All datasets are contaminated exclusively by Gaussian noise. Figure 3.34 shows the generated sets defined mathematically as

$$D1 : e = \mathcal{N}(0, 1)$$

$$D2 : e = 2 \sin\left(\frac{2\pi}{20}t\right) + \mathcal{N}(0, 1)$$

$$D3 : e = 2 \sin\left(\frac{2\pi}{20}t\right) + 5 \sin\left(\frac{2\pi}{200}t\right) + \mathcal{N}(0, 1)$$

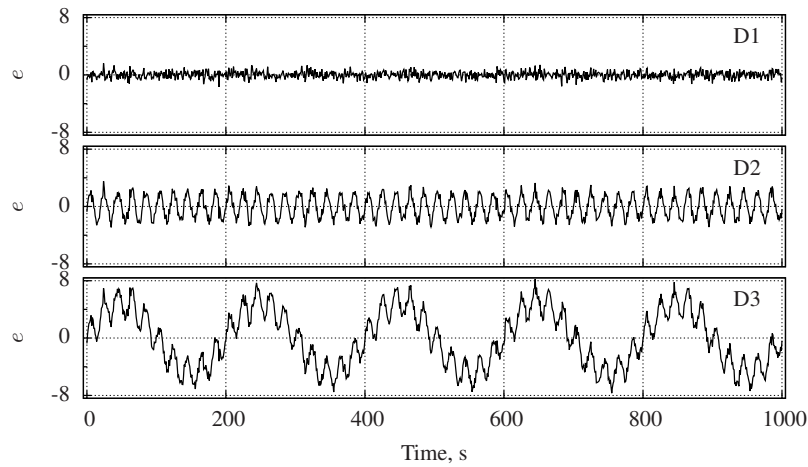


Figure 3.34: Simulated datasets.

**Industrial data:** Datasets collected in real environments (see Figure 3.35).

The first industrial dataset, D4, refers to a flow rate control loop affected by valve stiction that causes an oscillation of the signal and contains 1998 points. The points were collected with a sampling time of 10 s and the PID controller of the loop was tuned with a proportional gain  $k_C = 0.429 \% \text{kg}^{-1} \text{s}$  and an integral time  $\tau_I = 102 \text{ s}$ . This dataset, originally contributed by Scali, is publicly available in the database of Jelali and Huang (2013) where it is designated by CHEM32.

The second industrial dataset, D5, was collected between 2011 and 2012 in a large-scale production petrochemical plant from Group Sinopec Yangzi Petro-Chemical Co. (China). Provided by Wang, the dataset is available via LPMOM (2013) in file `Data_for_Example2.mat` and contains 12402 points.

The third dataset, D6, pertains to a level control loop and presents a sampling time of 5 s. It is constituted by 8641 points and is also available in the database of Jelali and Huang (2013) under the designation POW2. It was afforded by Choudhury.

Datasets D1 to D6 were independently analysed by the five methods A to E. Table 3.6 condenses the results obtained by each of the methods and also exposes,

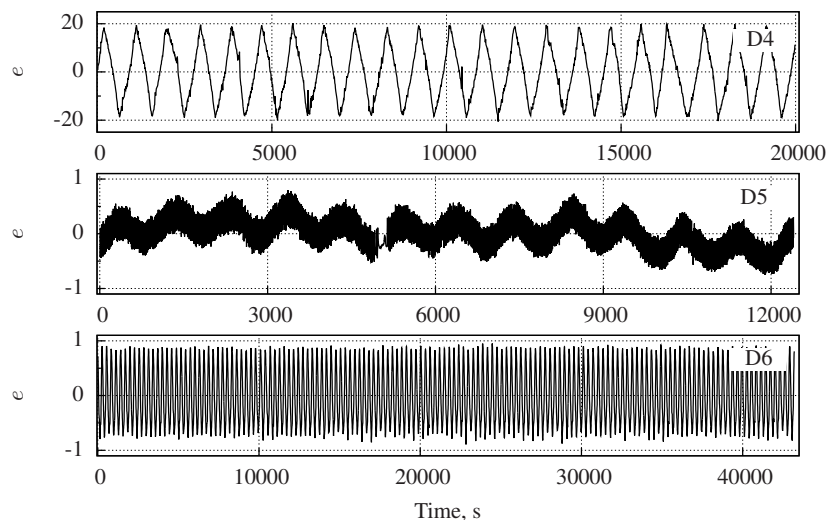


Figure 3.35: Industrial datasets.

in the last line, the true evaluation associated to each of the datasets in order to be possible to infer the degree of performance achieved by each method. While the characteristics of the signal were imposed in the case of the simulated datasets (and it is therefore easy to indicate the “true evaluation” characteristics), the industrial data “true evaluation” was obtained by visual inspection.

Table 3.6: Results of data analysis by different methods. The oscillation period(s) identified are indicated in seconds. The dark shadow indicates a wrong detection of the presence or absence of the oscillation(s) and the light shadow indicates a partially wrong or incomplete analysis.

	Simulated data			Industrial data		
	D1	D2	D3	D4	D5	D6
Method A	✓(4)	✓(20)	✓(95)	✓(794)	✓(18)	✓(294)
Method B	×	✓	×	✓	✓	✓
Method C	✓	✓	✓	✓	✓	✓
Method D	✓(15)	✓(20)	✓(20)	✓(300)	✓(14, 996)	✓(294)
Method E	×	✓(20)	✓(20, 200)	✓(898)	✓(14, 1010)	✓(294)
True Evaluation	×	✓(20)	✓(20, 200)	✓(≈ 900)	✓(≈ 15, 1000)	✓(≈ 300)

In order to express the results of the data analysis, a check symbol ✓ is used when the method detects the presence of oscillation(s) and a cross × otherwise. The check symbol ✓ is followed by the oscillation(s) characterization indicating in parenthesis its(their) period(s) whenever the method gives such information

(methods B and C do not characterize the detected oscillations). Similarly, for the line of the “true evaluation”, the cross symbol  $\times$  indicates the absence of oscillation, the check symbol  $\checkmark$  the presence and the number(s) in parenthesis is(are) the period(s) of oscillation.

### Method A

Method A detected oscillations disturbances in all the datasets, as listed along the first line of Table 3.6, even for the case of dataset D1 with which the method produced a wrong conclusion as this dataset belongs to a control loop performing well just affected by Gaussian noise. Moreover, it even suggests an oscillation period of 4 s to characterize the absent oscillation disturbance.

In opposition, for datasets D2 and D6 (which refer to datasets containing a single oscillation each) the method correctly identified the presence of the oscillation and also correctly characterized its period (exactly in the case of generated data D2 and approximately in the case of industrial data D6).

Since it is based on the zero crossings of the control error signal, the method leads to wrong detections when the controlled variable is not centered in the set-point variable. Besides, the presence of noise may skew the detection as the algorithm may interpret the  $x$ -axis crossings (provoked by noise) as the start or the end of a new oscillation. Although the minimum oscillation amplitude parameter,  $a$ , may reduce the noise impact, it must be defined individually which is not practicable when an automated solution is intended.

These issues also affected oscillation characterization in the case of the dataset D3, where the method proved unable to detect the presence of multiple oscillations and, moreover, the single period indicated by the method does not characterize any of the two existing oscillations. Also with the industrial dataset D5, method A showed to be unable to identify multiple oscillation disturbances.

As for dataset D4, the method detected the presence of the single existing oscillation but failed to characterize it, suggesting an oscillation period significantly different from the real one. Figure 3.36 depicts with more detail the results for this dataset. To detect and characterize oscillations, the  $IAE_{lim}$  was initially calculated as 32.47, using the value of  $\tau_1$ . As it is possible to see in Figure 3.36, the values of IAE rise frequently above the  $IAE_{lim}$  (which is recalculated in each iteration).



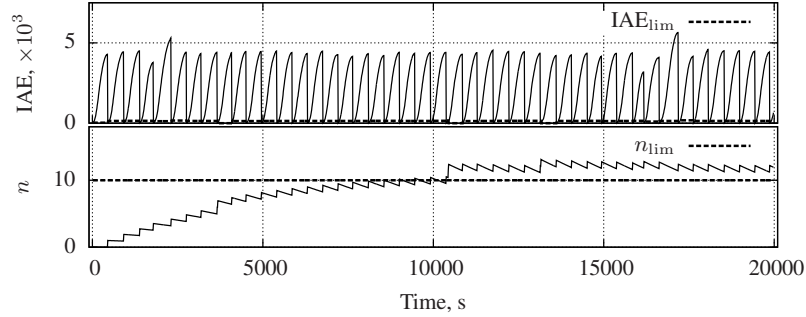


Figure 3.36: Application of method A to data D4.

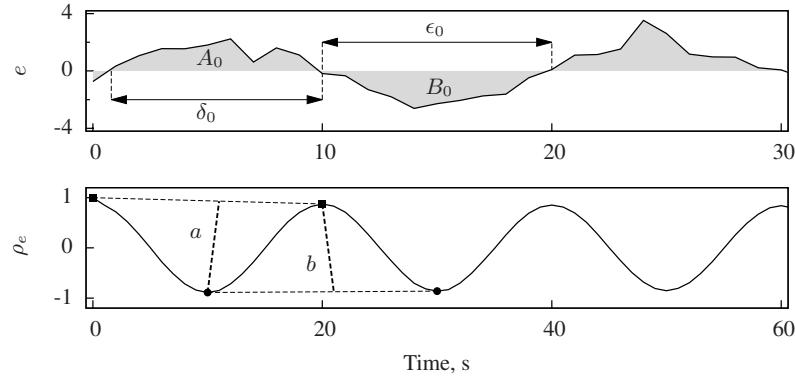


Figure 3.37: Graphical representation of the application of methods B (first plot) and C (second plot) to dataset D2. The apparent lack of orthogonality between the dashed line segments of the second plot is merely due to the different scales for abscissae and ordinates.

Each time the limit is touched, the disturbance number  $n$  increases, achieving quickly the maximum acceptable value approximately at 10000 s.

### Methods B and C

Methods B and C perform exclusively the oscillations detection task, being unable to compute the oscillation periods. Figure 3.37 highlights some variables associated to these methods (see (3.44) and (3.45) of method B and (3.50) of method C) considering the analysis of dataset D2.

Applying method B (see (3.48)),  $h_A = 39$  and  $h_B = 39$ , resulting an index of  $h = 0.639$ . According this method's criteria, such value indicates that the signal is possibly oscillating, being candidate for a closer examination.

Figure 3.37 also presents the auto-correlation function  $\rho$  of the control error

signal used by method C, emphasizing its capacity of signal filtering (compare, in terms of noise, to the signal in Figure 3.34). By applying method C, an index of  $R = 0.970$  is computed, which points to an excessive degree of oscillation of the signal.

Method B concluded correctly about the absence of oscillation disturbances for dataset D1 as well as about the presence of oscillations for datasets D2, D4, D5, and D6. However, it failed the analysis to dataset D3, wrongly indicating that the control loop of such data is working perfectly. In fact, this dataset is affected by oscillations characterized by two different oscillation periods. It is noteworthy that this method analyses directly the signal, what is potentially dangerous from the point of view of the results obtained. For instance, the signal can be not centered on the setpoint variable leading to wrong conclusions.

In contrast, method C identified correctly oscillations in D3 possibly due to the auto-correlation function usage which removed the signal noise and centered the signal around the  $x$ -axis. In spite of that, method C failed to recognize the oscillation absence in dataset D1.

### Method D

Method D wrongly indicated that the control loop of dataset D1 is affected by an oscillation disturbance. It also could not detect the multiple oscillations presence in dataset D3. Additionally, for dataset D4, the oscillation period with which the method characterizes the oscillation correctly detected is about 1/3 of its true value. However, it proved to perform very well for datasets D2, D5, and D6.

Analysing now in more detail the control error associated with dataset D5 (middle plot of Figure 3.35), it is possible to observe an oscillation of period around 1000 s and a second oscillation with a period around the 15 s (first plot of Figure 3.38, which consists of a zoom of Figure 3.35 for the first 195 s). The second plot of Figure 3.38 shows the discrete cosine transform analysed by method D. Several oscillations were successfully found by the application of the method filters. However, only two oscillations are considered dominant as only they have a regularity index greater than 1 ( $r = 21.0$  and  $r = 12.9$ ) and a high  $F$  value (13.1 and 10.7%). The two selected oscillations have periods of 996 s and 14 s. Dataset D5 was also studied by Wang et al. (2013). In their study, the authors

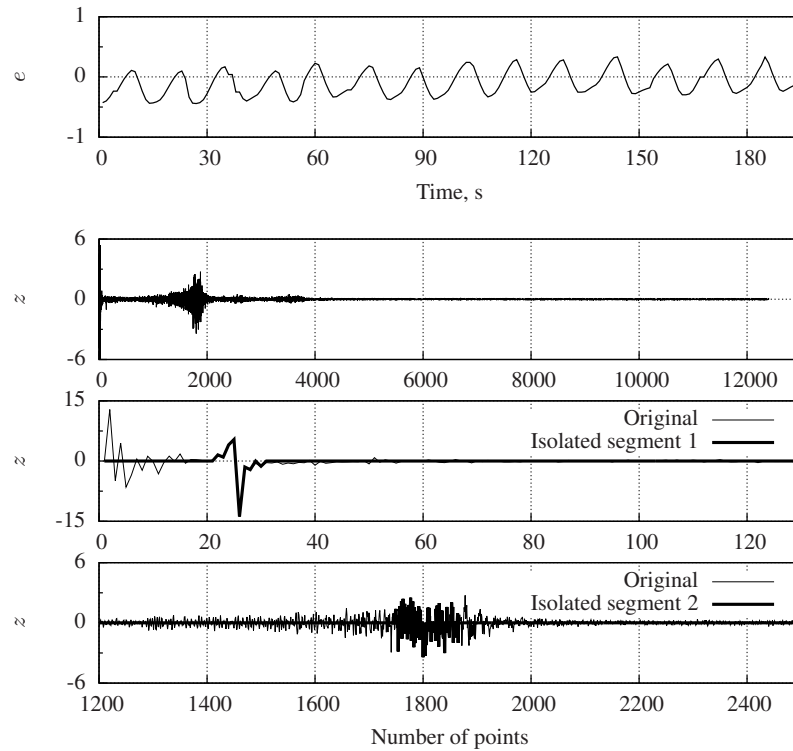


Figure 3.38: Application of method D to dataset D5. The first plot shows a zoom out of the time series D5, the second shows the signal of the discrete cosine transform, and the third and fourth plots show isolated transform segments for each of the oscillations detected.

found two disturbances oscillating with periods of 996 and 14 s characterized by the regularities 19.8 and 14.9 and indices  $F$  of 13.1 and 11.8%. These values are quite similar to those established in this study.

### Method E

As for the new method here proposed, it was able to extract correct conclusions with all the datasets under study: it was able to infer that the control loop of dataset D1 was working properly (absence of oscillation disturbances) while the loops of all the other datasets were affected by oscillation disturbances. It was able to correctly detect the cases of single and of multiple oscillations. Moreover, it computed successfully the oscillation periods for both single and multiple oscillations cases. It handled efficaciously both simulated and industrial datasets.

With exemplifying purposes, let's consider dataset D5 in more detail. The

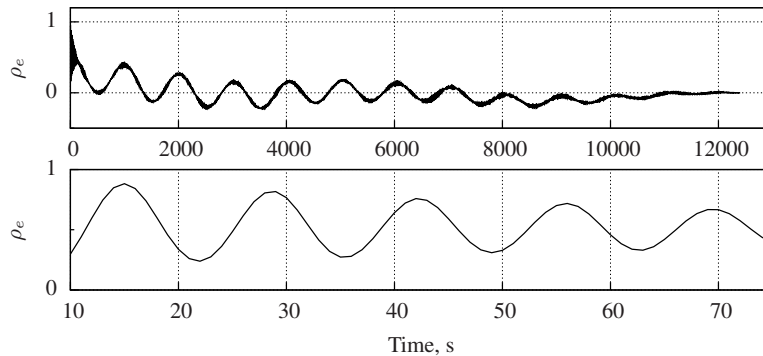


Figure 3.39: Auto-correlation function used by method E for analysis of dataset D5.

auto-correlation function was able to discern the different oscillations existing in that data: a more fast disturbance with oscillation period of 14 s (see second plot of Figure 3.39) and a slower oscillation of 1010 s (see first plot of Figure 3.39). The high values of the regularity indices (9.45 and 15.71) showed that these oscillations are regular which confirms their presence in the signal. The oscillations and their characteristics detected by method E for dataset D5 are similar to those determined by method D (two oscillations with periods 996 s and 14 s). Visual analysis confirms the existence of two oscillations with periods of approximately 1000 s and 15 s.

Besides dataset D5, dataset D3 also contains multiple oscillations. But for this dataset, method D was not able to detect both oscillations while method E kept its good performance.

The study accomplished showed that methods A, D, and E have the ability to detect only significant oscillations. In opposition to methods A and B, methods C, D, and E appear to be invulnerable to noise in the data. Methods D and E are the only ones that can determine multiple oscillations (however, method D has failed the detection of the multiple oscillations for dataset D3). As discussed above, the use of controlled variables or their associated control errors by the methods represents an advantage (presented by methods C and E). Besides determining the oscillation period, some of the methods also capable of characterizing its regularity, namely methods B, D, and E.

On the other hand, methods A and B require the signal to be centered in the

### 3.3. Detection and characterization of oscillating disturbances

Table 3.7: Advantages and disadvantages exhibited by oscillation detection and/or characterization methods

Advantages/disadvantages	Methods				
	A	B	C	D	E*
<i>Advantages</i>					
Detects only significant oscillations	●	○	○	●	●
Reduced impact of noise	○	○	●	●	●
Detects multiple oscillations	○	○	○	●	●
Uses controlled or control error variables	○	○	●	○	●
Characterizes the regularity of the found oscillation period	○	●	○	●	●
<i>Disadvantages</i>					
Requires that the signal is centered of the setpoint variable	●	●	○	○	○
Problems with zero crossings detection	●	●	○	●	○
Aplicable just to PID controllers with integral action	●	○	○	○	○
Problemas in determining several oscillation periods	●	●	●	●	○

\* Proposed approach.

setpoint variable and zero crossings detection are problematic. This last aspect is an issue to method D as well. Method A is the only one that can be applied only to PID controllers with integral action. Finally, all the methods except method E showed problems in determining the periods of multiple oscillations (either because the methods even do not detect the multiple oscillations themselves (cases of methods A, B, and C) or because, detecting them, they failed to always compute correctly the periods (method D)). This study showed that the new method E outperforms even the only other method that can detect multiple oscillations (method D).

Table 3.7 summarizes these advantages and disadvantages. A filled circle signifies that the corresponding characteristic is present for the corresponding method. A hollow circle signifies that the method does not present such characteristic.

The study here performed reveals the potential of the proposed approach when compared to previously existing methods. As the graphical description of Table 3.7 easily conveys, the new method offers simultaneously the advantages of all methods and, simultaneously, do not present any of their disadvantages.

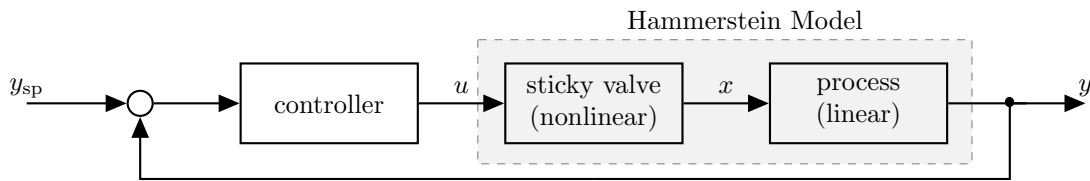


Figure 3.40: Industrial control loop representing the Hammerstein Model, where  $y_{sp}$  is the variable setpoint,  $u$  is the controller output,  $x$  is the real valve position, and  $y$  is the controlled variable.

### 3.4 Stiction detection and quantification through numerical optimization<sup>3</sup>

Stiction is one of the long-standing control valve problems in the process industry causing oscillations and, consequently, losses of productivity. Therefore, it is important to understand this phenomenon for its early detection and separation from other oscillation causes.

The present section explores the application of continuous optimization techniques to the system identification of a model with the stiction phenomenon. A strategy to deal with the discontinuities of the used Hammerstein Model in the context of the optimization procedure is proposed.

#### 3.4.1 Proposed approach

A novel technique for detection and quantification of valve stiction in control loops based on one-stage identification is proposed in this section. The system to be identified is represented by the Hammerstein Model shown in Figure 3.40. It consists of a static non-linear element in series with a linear dynamic part (Eskinat et al., 1991). In the context of an industrial control loop, the non-linear element represents the sticky valve while the linear part models the process dynamics.

<sup>3</sup>This section is based on Brásio, A. S. R., Romanenko, A., and Fernandes, N. C. P. (2014). Stiction detection and quantification as an application of optimization. In Murgante, B., Misra, S., Rocha, A., Torre, C., Rocha, J., Falcão, M., Tanir, D., Aduhan, B., and Gervasi, O., editors, *Computational Science and Its Applications – ICCSA 2014*, volume 8580 of *Lecture Notes in Computer Science*, pages 169–179. Springer International Publishing. URL [http://dx.doi.org/10.1007/978-3-319-09129-7\\_13](http://dx.doi.org/10.1007/978-3-319-09129-7_13), with kind permission from Springer Science and Business Media B. V.

### 3.4. Stiction detection and quantification through numerical optimization

After sufficiently rich process data is collected, the model parameters are determined such that the model response reproduces the observed response of the actual process. Mathematically, this is represented by the non-linear constrained optimization problem

$$\underset{\mathbf{p}}{\text{minimize}} \quad J(\mathbf{y}, \mathbf{u}, \mathbf{p}) \quad (3.112a)$$

$$\text{subject to} \quad \dot{\mathbf{y}} = f(\mathbf{y}, \mathbf{x}, \mathbf{p}) \quad (3.112b)$$

$$\mathbf{x} = g(\mathbf{u}, \mathbf{p}) \quad (3.112c)$$

$$\mathbf{y}_L \leq \mathbf{y} \leq \mathbf{y}_U \quad (3.112d)$$

$$\mathbf{u}_L \leq \mathbf{u} \leq \mathbf{u}_U \quad (3.112e)$$

$$\mathbf{p}_L \leq \mathbf{p} \leq \mathbf{p}_U \quad (3.112f)$$

$$h(\mathbf{p}) \leq 0, \quad (3.112g)$$

where  $J$  denotes the objective function,  $\mathbf{p}$  is the parameters vector including both the stiction and the process models,  $\mathbf{y}$  and  $\mathbf{u}$  are the vectors of controlled variable and controller output (respectively),  $\mathbf{x}$  is the vector of the real valve position, and the subscripts  $_L$  and  $_U$  stand for lower and upper bounds (respectively). The set of equations (3.112b) and (3.112c) defines a set of constraints arising from the Hammerstein Model dynamics. Inequalities (3.112g) enforce additional identification criteria.

As it may be seen in Figure 3.40, the non-linear element scales the controller output and transforms it to the real valve position. The model expressed by the set of equations (3.112c) corresponds to this transformation and is represented by a stiction model existent in the literature. In contrast, the linear element whose output is the controlled variable is modeled by the linear model specified by (3.112b).

Given the Hammerstein Model and a set of  $n$  experimental data points  $(t_i, y_{\text{exp},i})$ , the objective function  $J$  is written, according to the minimum least squares criterion, as

$$J = [\mathbf{y}_{\text{exp}} - \mathbf{y}]^T \mathbf{Q} [\mathbf{y}_{\text{exp}} - \mathbf{y}], \quad (3.113)$$

where  $\mathbf{Q}$  is a diagonal matrix containing the weights given to each observed variable. In this work, equal weight was given to all output variables.

From a practical point of view, the proposed technique only requires the controller output and the controlled variable data that may be accessed in the DCS (Distributed Control System) of industrial plants. Notice that the real valve position is an unmeasured intermediate variable, but the method does not require it.

### 3.4.2 Smoothing of discontinuous models

Stiction is essentially described by discontinuous non-linear models and that calls for mixed integer non-linear optimization problem formulation and a special class of optimizers. Alternatively, smoothing approaches for discontinuous models have been successfully applied. Some works introduced smoothing techniques in the context of exact penalty functions (Wu et al., 2004, 2005; Meng et al., 2009). Others authors have suggested to express discontinuities by means of a step function and then to substitute this function by a continuous approximation (Goldfeld and Quandt, 1972; Tishler and Zang, 1979; Zang, 1981). This is the approach also adopted in the present work.

Consider the general discontinuous system

$$z(t) = \begin{cases} z_1(t), & \text{if } t \in T_1 \\ z_2(t), & \text{if } t \in T_2 \\ \vdots \\ z_m(t), & \text{if } t \in T_m \end{cases}, \quad (3.114)$$

where  $z_i(t)$ ,  $i = 1, \dots, m$ , are continuously differentiable real functions over  $\mathbb{R}^n$  subject to the conditions that define the subsets  $T_i$ . Assuming that the real expressions  $e_k(t)$ ,  $k = 1, \dots, p$ , are continuously differentiable over  $\mathbb{R}^n$ , the subsets  $T_i$  are defined as

$$T_i = \{t \in \mathbb{R}^n : e_k(t) < 0, \forall k \in L_i; e_k(t) \geq 0, \forall k \in G_i\}, \quad (3.115)$$

where  $L_i$  and  $G_i$  are, for branch  $i$ , the sets of indexes  $k$  for which  $e_k(t) < 0$  and  $e_k(t) \geq 0$ , respectively.

The discontinuous function (3.114) may be expressed by means of the Heavi-



side function as

$$z(t) = \sum_{i=1}^m \prod_{k \in L_i} [1 - \mathcal{H}(e_k)] \prod_{k \in G_i} \mathcal{H}(e_k) z_i(t), \quad (3.116)$$

with

$$\mathcal{H}(t) = \begin{cases} 1, & \text{if } t \geq 0 \\ 0, & \text{if } t < 0 \end{cases}, \quad (3.117)$$

that is,

$$\mathcal{H}(e_k) = \begin{cases} 1, & \text{if } e_k \geq 0 \\ 0, & \text{if } e_k < 0 \end{cases}. \quad (3.118)$$

It is possible to smooth the Heaviside function by approximating it by the hyperbolic function

$$\tilde{\mathcal{H}}(t) = 0.5 + 0.5 \cdot \tanh(r \cdot t), \quad (3.119)$$

where  $\tilde{\mathcal{H}}(t)$  is second-order continuously differentiable on  $\mathbb{R}^n$  varying within the interval  $[0, 1]$ , and  $r$  is an accuracy parameter. Similarly to the approach of Zang (1981), the step function approximation here considered contains a single parameter. This parameter controls the accuracy of the approximation by adjusting the size of the neighborhoods around the discontinuity points over which the approximation has an effective effect.

Therefore, the continuous differentiable on  $\mathbb{R}^n$  function that approximates function (3.114) may be written as

$$\tilde{z}(t) = \sum_{i=1}^m \prod_{k \in L_i} [1 - \tilde{\mathcal{H}}(e_k)] \prod_{k \in G_i} \tilde{\mathcal{H}}(e_k) z_i(t). \quad (3.120)$$

### 3.4.3 Development for a system containing a sticky valve

As mentioned above, the Hammerstein Model comprises a non-linear model describing the sticky valve and a linear process model. The present work uses the complete Chen Model, also called by its authors as two-layer binary tree model (Chen et al., 2008), to model the sticky valve.

In what concerns the process dynamics, it is modeled by the SISO state-space

model

$$\dot{\mathbf{y}}^* = a \mathbf{y}^* + b \mathbf{x}^*, \quad (3.121)$$

where  $a$  and  $b$  are state-space model constants, and the deviation variables vectors  $\mathbf{y}^*$  and  $\mathbf{x}^*$  are related to the original variables  $\mathbf{y}$  and  $\mathbf{x}$  through the simple translations  $\mathbf{y}^* = \mathbf{y} - \bar{\mathbf{y}}$  and  $\mathbf{x}^* = \mathbf{x} - \bar{\mathbf{x}}$ , respectively.

In order to collect experimental data needed to perform a comparison between the smoothed and the original versions of the Chen Model and also needed for the identification process, a plant simulation was carried out using the Hammerstein Model containing the original discontinuous Chen Model. The parameters used in the simulation are: (i) for the stiction model:  $f_S = 2.8\%$  and  $f_D = 0.9\%$ ; (ii) for the process model:  $a = 1$ ,  $b = -1\%^{-1}$ ,  $\bar{y} = 0$ , and  $\bar{x} = 0\%$ . A sinusoidal excitation on the controller output with amplitude of 5% and period of 40min is applied to the system to generate a response,  $\mathbf{y}_{\text{exp}}$ , that includes sufficient dynamical information about the valve and the process. The obtained dataset contains  $n = 101$  points covering an interval of 50 minutes with a sampling period of 0.5 minutes.

### ■ Smoothing of the stiction model

An enhanced flow diagram of the Chen Model was built and is illustrated in Figure 3.41. The diagram is complemented, relatively to the original model presented by its authors (Chen et al., 2008), with some notes to better explain the model and the approach developed in the present thesis.

The Chen Model may be rewritten as

$$z(t) = \begin{cases} z_1(t), & \text{if } e_1(t) = 1 \wedge e_2(t) \geq 0 \\ z_2(t), & \text{if } e_1(t) = 1 \wedge e_2(t) < 0 \\ z_3(t), & \text{if } e_1(t) = 0 \wedge e_3(t) < 0 \\ z_4(t), & \text{if } e_1(t) = 0 \wedge e_3(t) \geq 0 \end{cases}, \quad (3.122)$$

where  $z(t)$  is a general variable used to represent the outputs  $c(t)$ ,  $x(t)$  and  $s(t)$ .

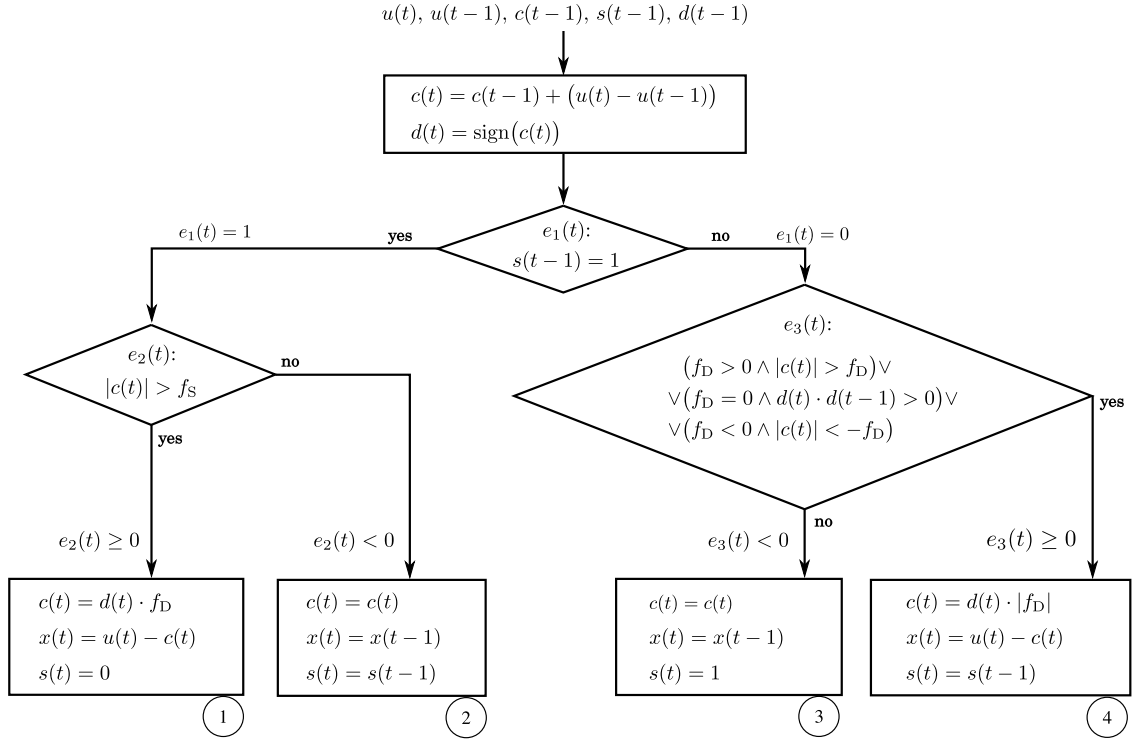


Figure 3.41: Enhanced Chen Model flow diagram.

Expressions  $e_1(t)$  and  $e_2(t)$  are given by

$$e_1(t) = s(t-1), \quad (3.123)$$

$$e_2(t) = c(t) - f_S, \quad (3.124)$$

where  $s(t)$  is the valve status flag,  $c(t)$  is the accumulated force compensated by friction, and  $f_S$  is a model constant. The expression  $e_3(t)$  becomes positive or equal to zero when

$$\left( e_{31}(t) \geq 0 \wedge e_{32}(t) \geq 0 \right) \vee \left( e_{31}(t) < 0 \wedge e_{33}(t) \geq 0 \right), \quad (3.125)$$

with

$$e_{31}(t) = f_D, \quad (3.126)$$

$$e_{32}(t) = |c(t)| - f_D, \quad (3.127)$$

$$e_{33}(t) = d(t) \cdot d(t-1), \quad (3.128)$$

where  $d(t)$  is the movement direction, and  $f_D$  is a model constant. Notice that the condition  $f_D < 0 \wedge |c(t)| < -f_D$  (see decision diamond-shaped box  $e_3(t)$  of Figure 3.41) is not considered in the approach, because it is assumed that  $f_D \geq 0$ . The expression  $e_3(t)$  becomes negative otherwise.

The Chen Model contains  $m = 4$  branches subject to  $p = 3$  conditions. Sets  $L_i$  and  $G_i$  are defined as

$$\begin{aligned} L_1 &= \{\}, & L_2 &= \{2\}, & L_3 &= \{1, 3\}, & L_4 &= \{1\}, \\ G_1 &= \{1, 2\}, & G_2 &= \{1\}, & G_3 &= \{\}, & G_4 &= \{3\}. \end{aligned}$$

The continuous and differentiable function that approximates (3.122) is therefore defined by

$$\begin{aligned} \tilde{z}(t) &= e_1(t) \cdot \tilde{\mathcal{H}}(e_2) \cdot z_1(t) + [1 - \tilde{\mathcal{H}}(e_2)] \cdot e_1(t) \cdot z_2(t) \\ &+ [1 - e_1(t)] \cdot [1 - \tilde{\mathcal{H}}(e_3)] \cdot z_3(t) + [1 - e_1(t)] \cdot \tilde{\mathcal{H}}(e_3) \cdot z_4(t), \end{aligned} \quad (3.129)$$

with

$$\tilde{\mathcal{H}}(e_3) = \tilde{\mathcal{H}}(e_{31}) \cdot \tilde{\mathcal{H}}(e_{32}) + [1 - \tilde{\mathcal{H}}(e_{31})] \cdot \tilde{\mathcal{H}}(e_{33}). \quad (3.130)$$

It is noteworthy that  $e_1(t)$  is used in (3.122) inside an equality condition which precludes the direct usage of the approximation approach described in Section 3.4.2. However, as shown in (3.123), this condition is given by the output  $s(t-1)$  which is smoothed and valued between 0 and 1 similarly to  $\tilde{\mathcal{H}}(t)$ . These facts enable the use of this variable directly in equation (3.129) in a similar way as the smoothed Heaviside functions, allowing to deal with the equality constraint.

Several simulations were performed in order to assess the performance of the developed approach. Figure 3.42 depicts the simulation responses of the Hammerstein Model when the non-linear element is described by the original Chen Model (solid line) and also when the non-linear element is described by the smoothed Chen Model, proposed in the present thesis (dashed lines and points). Being a measure of the quality of the approximation applied, the parameter  $r$  has a visible influence on the performance of the smoothed model. This influence is quantified in Table 3.8 by the mean squared error (MSE) associated with the

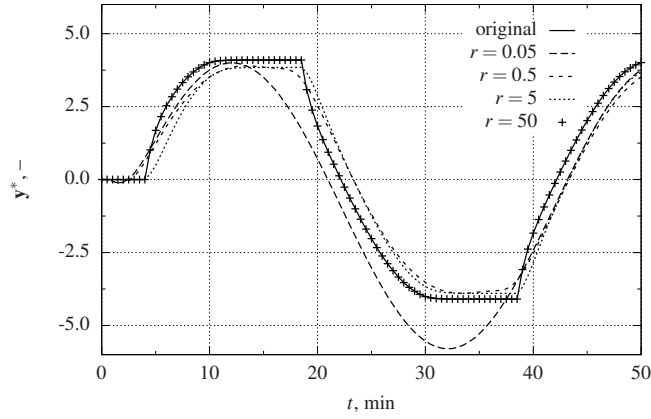


Figure 3.42: Comparison between the Hammerstein Model using the original Chen Model and its smoothed version for different values of  $r$ .

Table 3.8: MSE for different values of  $r$ .

$r$	0.05	0.5	5	10	20	30	40	50
MSE	88.440	34.459	39.701	0.829	0.011	0.003	0.002	0.000

simulations for different values of  $r$ . As it may be easily seen in Figure 3.42, by using bigger values of  $r$  it is possible to reproduce better the data obtained by the original Chen Model. For bigger values of  $r$ , the approximation of the function occurs in smaller neighborhoods of the discontinuity points leading to a better approximation. The value of  $r = 50$  was selected based on a mean squared error tolerance of  $10^{-3}$ .

### ■ Stiction detection and quantification

The Chen Model parameter  $f_S$  is linearly dependent on  $f_D$  through the mathematical relationship (Chen et al., 2008)

$$f_S = f_D + f_J. \quad (3.131)$$

Such dependence poses a difficulty in system identification, because it compromises the identifiability of the individual parameters. In order to overcome this problem, the model is reformulated to use parameters  $f_J$  instead of  $f_S$  for optimization purposes.

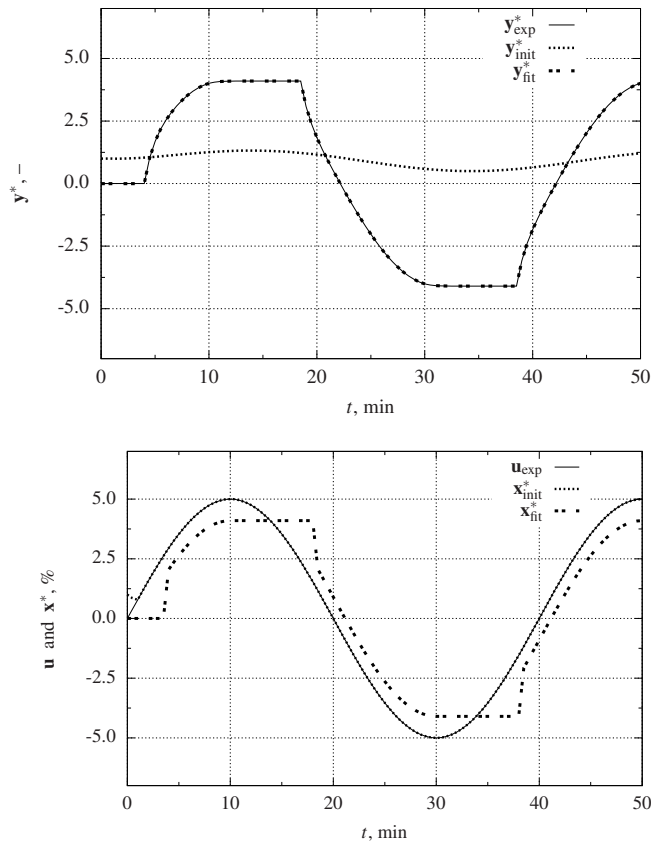


Figure 3.43: System identification.

The optimization procedure described in Section 3.4.1 was used to identify the system. The implementation was made in GNU Octave 3.6.3 using its general non-linear `sqp()` (successive quadratic programming) solver. The set of optimization related conditions and the obtained model parameters as well as some fitting quality indicators are presented in Table 3.9. The optimization tolerance was  $10^{-20}$ . In order to avoid poor conditioning of the data, the parameters were normalized by the vector  $\alpha = [-0.1 \ 0.1 \ 1 \ 1 \ 1 \ 1]$ , with the necessary changes in the model.

The profile predicted by the identified model,  $y_{fit}^*$ , may be directly compared with the experimental profile,  $y_{exp}^*$ , in Figure 3.43. The initial profiles,  $y_{init}^*$  and  $x_{init}^*$ , are also displayed revealing that the starting initial situation was significantly different from the experimental profiles. As it is possible to observe, the fitted Hammerstein Model is able to capture well the sticky valve and the process dynamics. The high correlation factor,  $R^2$ , and the lower objective function prove

Table 3.9: System identification results.

$\mathbf{p}$	Initial	$\mathbf{p}_L$	$\mathbf{p}_U$	Fit	Indicators
$a, -$	-0.200	-10	10	-0.998	
$b, \%$	0.020	-10	10	0.998	
$\bar{y}, -$	1.000	-10	10	0.000	$J = 0.001$
$\bar{x}, \%^{-1}$	1.000	-10	10	0.000	$R^2 = 1.000$
$f_J, \%$	0.000	0	10	1.732	
$f_D, \%$	0.000	0	10	0.898	

$\mathbf{Q} = 10^2 \mathbf{I}_{101}$ , where  $\mathbf{I}_n$  is the identity matrix of size  $n \times n$

the effectiveness of the one-stage system identification technique.

### 3.5 Detection of stiction in level control loops<sup>4</sup>

Stiction is an enduring problem of control loops in process industry. When it occurs, the real position of the valve stem can differ substantially from the controller output (see Figure 3.44) deteriorating the performance of the control loop.

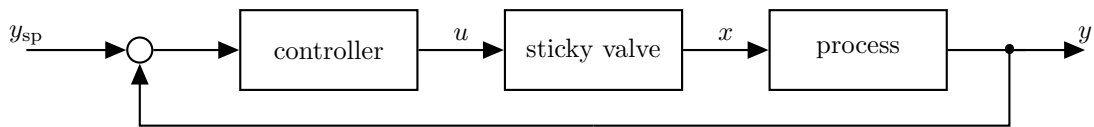


Figure 3.44: Industrial control loop with stiction, where  $y_{sp}$  is the variable set-point,  $u$  is the controller output,  $x$  is the real valve position, and  $y$  is the controlled variable.

By applying Yamashita's method (Yamashita, 2006a) to a considerable number of industrial flow control loops, Manum and Scali (2006) concluded that it diagnoses the presence of stiction in half of the occurrences. However, this method presents the disadvantage of requiring valve stem position data. Even though this data is often unavailable, it is nevertheless possible to apply the method in

<sup>4</sup>This section is based on Brásio, A. S., Romanenko, A., and Fernandes, N. C. (2015). Detection of stiction in level control loops. *IFAC-PapersOnLine*, 48(8):421 – 426. 9th IFAC Symposium on Advanced Control of Chemical Processes ADCHEM 2015 Whistler, Canada, June 7-10, 2015. © IFAC 2015. Reproduced with the permission of IFAC. The original version was published in IFAC-PapersOnline and can be found using the Digital Object Identifier 10.1016/j.ifacol.2015.09.004.



Figure 3.45: Typical patterns of a sticky valve.

flow control loops with the assumption of linearity and fast dynamics. Indeed, in such case the controlled variable is proportional to the real valve position. Later, that disadvantage was addressed (Yamashita, 2006b) by developing a new index for systems with slower dynamics, namely level control loops, based on the detection of a two-peak distribution in the signal. However, this approach tends to produce false positive stiction detection, which undermines the method credibility.

The present work develops a new approach to detect valve stiction in level control loops that is based on the preprocessing of the variable profiles prior to the application of the pattern recognition of Yamashita (2006b).

### 3.5.1 Yamashita's method

Yamashita's method is designed for control loops with pneumatic actuators. The algorithm is based on the qualitative description of the changes suffered by the signals to and from the valve and showed excellent performance in the detection (Yamashita, 2006a).

Yamashita's method describes the typical patterns in the graphical representation of the real valve position versus the controller output ( $x$ - $u$  phase plot) associated with the stem movement. Figure 3.45 shows those idealized typical patterns of a sticky valve.

The qualitative changes of a signal may be represented using a sequence composed by the symbolic values **I**, **S**, **D** meaning increasing, steady and decreasing, respectively, and are represented in Figure 3.46 (top). The identification of the symbols is based on the time derivatives of the signals for each sampling point. For instance, at a given sampling point where the signal  $u$  increases while the signal  $x$  is steady, the symbolic representation is **IS**. For detecting stiction, Yamashita's method uses two main indexes:  $\rho_1$  and  $\rho_3$ . The index  $\rho_1$  counts the periods of sticky movements by finding **IS** and **DS** shapes in the phase plot. The



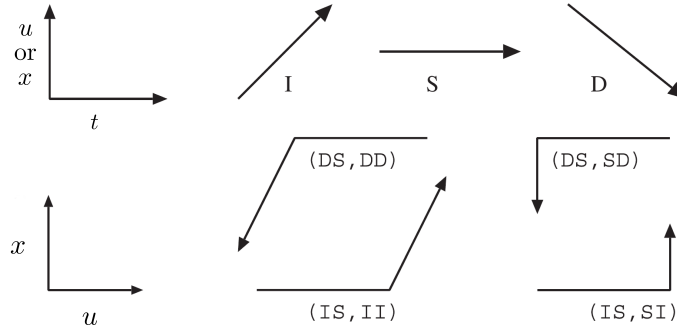


Figure 3.46: Symbols used to represent a signal (top) and typical qualitative shapes found in sticky valves (bottom).

index  $\rho_3$  takes into account the fact that some fragments of the stiction patterns may be represented by several sequences of two shapes (**IS II**, **DS DD**, ... as shown in Figure 3.46 (bottom)). Those indexes are calculated by

$$\rho_1 = \frac{\tau_{IS} + \tau_{DS}}{\tau_{total} - \tau_{SS}}, \quad (3.132)$$

$$\rho_3 = \rho_1 - \frac{\tau_{IS\ DD} + \tau_{IS\ DI} + \tau_{IS\ SD} + \tau_{IS\ ID} + \tau_{IS\ DS}}{\tau_{total} - \tau_{SS}} + \frac{\tau_{DS\ DI} + \tau_{DS\ SI} + \tau_{DS\ ID} + \tau_{DS\ II} + \tau_{DS\ IS}}{\tau_{total} - \tau_{SS}}, \quad (3.133)$$

where  $\tau_{total}$  is the width of the time window and  $\tau_p$  is the time periods for pattern  $p$  (with  $p = IS, IS\ DD, \dots$ ). Varying between 0 and 1, these indexes get higher if the valve has severe stiction. The authors inferred that the loop is likely to have valve stiction if the index values are greater than 0.25.

Later, Yamashita (2006b) developed a new index for systems with slower dynamics based on the detection of a two-peak distribution in the signal. It is based on the idea that the distribution of the difference between consecutive level measurements contains two separate peaks. To monitor valve stiction, the author uses the excess kurtosis statistical index to verify the distribution peaks. The excess kurtosis is defined as

$$\gamma = \frac{1}{n} \sum_{i=1}^n \frac{(\Delta y_i - \mu_{\Delta y})^4}{\sigma_{\Delta y}^4} - 3, \quad (3.134)$$

where  $\Delta y$  is the differential of  $y$ ,  $\mu_{\Delta y}$  and  $\sigma_{\Delta y}$  are the mean and the standard deviation of  $\Delta y$ , and  $n$  is the number of observations of  $\Delta y$ . A loop with stiction

will present a two peaked distribution which means a negative large value of excess kurtosis.

### 3.5.2 Proposed approach

The amount of the liquid stored in a vessel may be found by measuring the level of the liquid,  $y$ . The dynamics of a container filled with liquid is defined through the mass balance for constant density,  $\rho$ , and constant cross-sectional area,  $A$ , of the container as

$$\rho A \frac{dy}{dt} = F_{in} - F_{out} , \quad (3.135)$$

where  $F_{in}$  and  $F_{out}$  are the input and output mass flow rates, respectively. Considering linear installed flow characteristic  $F = a x$ , the balance shows that the valve position is directly proportional to the time variation of the vessel level, that is,

$$\frac{dy}{dt} \propto x . \quad (3.136)$$

As mentioned above, Yamashita's method performs well in flow rate control loops because it assumes that the controlled variable  $y$  is almost proportional to the real valve position  $x$ . However, such assumption is not valid for level loops and Yamashita's method fails because the dynamic patterns are different from those expected in flow control loops.

The rationale behind the present approach consists in applying a transformation function to the data to obtain a direct relation to the real valve position and only then apply the well-known Yamashita's method. Different transformation is required for self-regulating and integrating processes. In the later, which is the subject of this work, the transformation function  $f(y)$  is defined by (3.136) using the finite difference approximation

$$f(y) = \frac{y(t+1) - y(t)}{\Delta t} , \quad (3.137)$$

where  $\Delta t$  is the sampling time.

Figure 3.47 shows the real valve position  $x$  (first row) and the controlled variable  $y$  (second row) from a simulated level control loop containing a healthy valve (left column) and a sticky valve (right column). The application of the transfor-

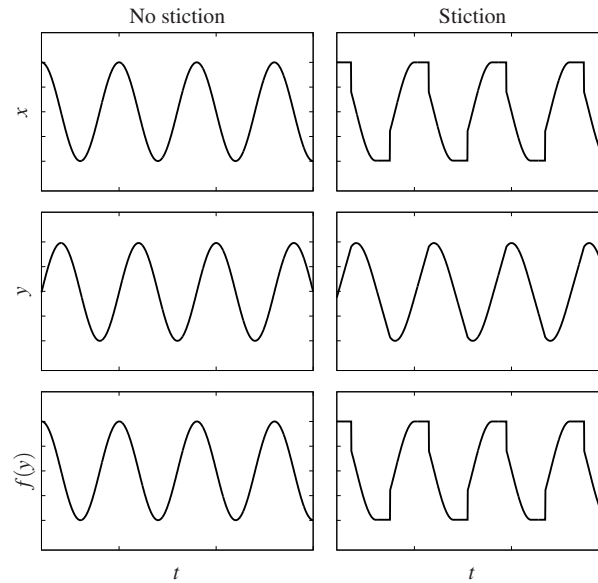


Figure 3.47: Real valve position  $x$ , controlled variable  $y$  and transformation function  $f(y)$  applied to the level for no stiction and stiction cases.

mation function  $f(y)$  to the level data is also drawn in the same figure (third row) showing how similar the transformed signal becomes to the real valve position for both cases.

Although this extension is only applicable to level control loops data, it merely uses operational data easily available in plants (the controller output  $u$  and the controlled variable  $y$ ) and requires no parameter tuning.

### 3.5.3 Application to a simulated system

This section presents an evaluation of the proposed approach using simulated datasets generated by an Hammerstein Model which is frequently used to model the stiction phenomenon. As already mentioned earlier, the Hammerstein Model consists of a non-linear element in series with a linear dynamic part. In the present context, the non-linear element represents the sticky valve while the linear part models the process dynamics. The present application uses the Choudhury Model to model stiction and the state-space model

$$\dot{y}(t) = a y(t) + b x(t), \quad (3.138)$$

where  $a$  and  $b$  are state-space model constants, to model the process dynamics.

In order to collect the experimental data, a plant simulation was carried out using the defined Hammerstein Model and the control algorithm

$$u(t) = u(t - 1) + k_C e(t) + \frac{1}{\tau_I} \int_0^t e(t) dt + \tau_D \dot{e}(t), \quad (3.139)$$

where  $e(t)$  is the error signal,  $k_C$  the proportional gain,  $\tau_I$  the integral time (or reset time), and  $\tau_D$  the derivative time. Model parameters in Choudhury et al. (2005) were used to generate data of a level control loop:  $a = 0 \text{ min}$ ,  $b = 1 \text{ m } \%^{-1}$ ,  $k_C = 0.4 \% \text{ m}^{-1}$ ,  $\tau_I = 0.2 \text{ min}^{-1}$ , and  $\tau_D = 0 \text{ min}$ . The Choudhury Model parameters  $(S, J)$  were defined as:  $(0, 0) \%$  for no stiction,  $(3, 0) \%$  for pure deadband,  $(3, 1.5) \%$  for stiction with undershoot, and  $(3, 3) \%$  for stiction with no-offset. Figure 3.48 and 3.49 show the collected data. Both are composed by two parts showing the time trends of  $u$  versus  $x$  (part (a)) and  $u$  versus  $y$  (part (b)). Each of these parts (a) and (b) is constituted by two columns showing the signals time trends at the left-hand and the corresponding phase plots at the right-hand. It is noteworthy that only  $u$  and  $y$  data are usually available from plants.

Figures 3.48 and 3.49 show the collected data. Both are composed by two parts showing the time trends of  $u$  versus  $x$  (part (a)) and  $u$  versus  $y$  (part (b)). Each of these parts (a) and (b) is constituted by two columns showing the signals time trends at the left-hand and the corresponding phase plots at the right-hand. It is noteworthy that only  $u$  and  $y$  data are usually available from plants.

Figure 3.48 represents the open-loop simulated response when the valve is driven by a sinusoidal variation in the input variable  $u$ . The first row shows a healthy valve (no stiction) where the real valve position  $x$  follows the input  $u$ . The second row exemplifies the pure deadband case. Here the valve sticks during a percentage of valve opening corresponding to the parameter  $S$ . The third, fourth and fifth rows represent cases of stiction with undershoot, no-offset, and overshoot, respectively. In accordance, the  $u$ - $x$  phase plots of the three cases in Figure 3.48a show clearly the presence of stiction. However, the respective  $u$ - $y$  phase plots in Figure 3.48b (which is the available data in plants) shows elliptical loops with sharp turn around which are not a very reliable shape to detect stiction.

Figure 3.49 illustrates the closed-loop response to a step in the variable set-

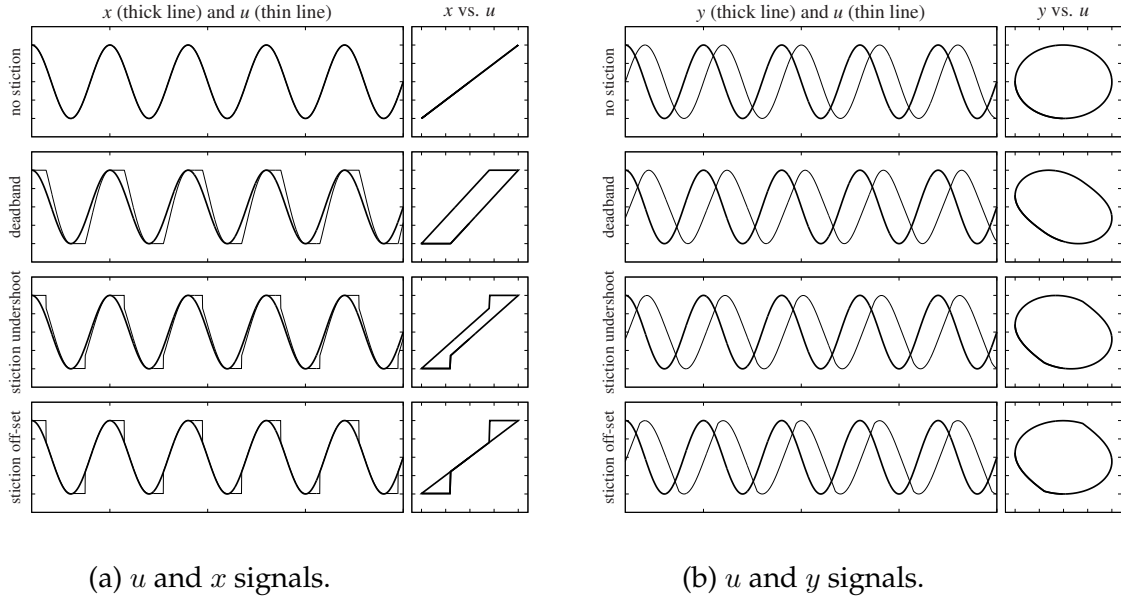


Figure 3.48: Open-loop response of a level control loop obtained by simulation using the Choudhury Model.

Table 3.10: Stiction detection results for free-noise closed-loop data of the level control. The shadow indicates a wrong detection.

Case	True Eval.	Yamashita's Method			Yamashita's index for slower dynamics		New Approach		
		$\rho_1$	$\rho_3$	Eval.	$\gamma$	Eval.	$\rho_1$	$\rho_3$	Eval.
No stiction	×	0.22	0.22	×	48.87	×	0.05	0.05	×
Pure deadband	✓	0.20	0.20	×	-1.80	✓	0.48	0.48	✓
Stiction undershoot	✓	0.02	0.02	×	-1.97	✓	0.96	0.93	✓
Stiction no-offset	✓	0.27	0.25	✓	-1.99	✓	0.95	0.90	✓

point  $y_{sp}$ . If stiction is present, the behavior of the control loop deteriorates producing unwanted limit cycles in the real valve position  $x$  and, therefore, in the controlled variable  $y$ . The third, fourth and fifth rows of Figure 3.49a and 3.49b clearly exhibit these cycles. The second row evidences that an integrator produces limit cycles even in the presence of pure deadband.

The approach developed in the present work was applied to the generated closed-loop data. The transformation function  $f(y)$  was calculated using (3.137) for the level data  $y$ . Then, Yamashita's method was applied to the variable  $u$  and to the transformed signal  $f(y)$ . Table 3.10 presents the numerical results for all the datasets, under the reference "New Approach".

The expected evaluation for detection of stiction is pointed out in the second

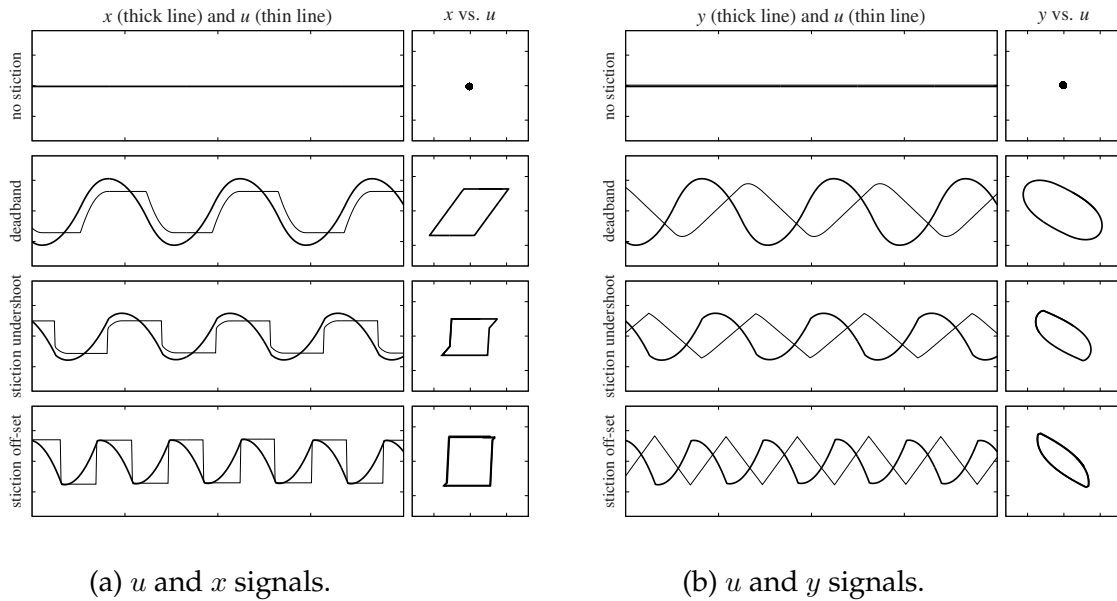


Figure 3.49: Closed-loop response of a level control loop obtained by simulation using the Choudhury Model.

column.

With comparison purposes, two other techniques were applied to the same datasets. Yamashita’s original method was applied using variable  $u$  and the controlled variable  $y$ . The study was complemented with the results of the version of Yamashita’s method for slower dynamics (Yamashita, 2006b). The later was applied using just the controlled variable  $y$ . The results of these two techniques are also shown in Table 3.10.

The performance evaluation of the methods on the simulated noise free closed-loop data (shown in Figure 3.49) reveals that Yamashita’s method produces two wrong detections in the cases of deadband and stiction with undershoot whereas Yamashita’s index for slower dynamics detects correctly the stiction phenomenon for the four studied cases. The results of the new approach proposed in this work are also correct and consistent for all the cases.

It is worth emphasising that such results were obtained for noise-free simulated data, which is uncommon in real industrial practice.

### 3.5.4 Influence of noise in the detection

The presence of noise in industrial data greatly impacts the plant performance analysis as it may obfuscate relevant information and, consequently, affect the algorithms. In this section, the influence of noise on the performance of the proposed stiction detection approach as well as on the performance of the other two techniques is studied. At first, the performance of the three methods was scrutinized by analysing how they handled sets of simulated data adulterated by noise. Moreover, different intensities of noise were studied. Finally, the three methods were compared when dealing with industrial data.

The dataset undergoes filter and downsampling as follows. The generated dataset is subdivided in 10 datapoint windows and a straight line is fitted within each of the intervals using the least-square criterion. The obtained function is used to calculate the value at the beginning of the interval.

#### ■ Using simulated data

Noisy closed-loop data was generated with the parameters mentioned above and with several degrees of noise  $n_i$  added to the controlled variable. The results of the detection methods are presented in Table 3.11 where the characterization of the added noise is also explicitly defined.

The presence and intensity of noise degrades the performance of Yamashita's original method and, especially, of Yamashita's index for slower dynamics. In the presence of noise, both methods give false positives and the second method additionally gives false negatives when the noise is more intense. In opposition, the proposed method produced the expected diagnosis results for all the cases highlighting its capacity to detect stiction even in noisy environments.

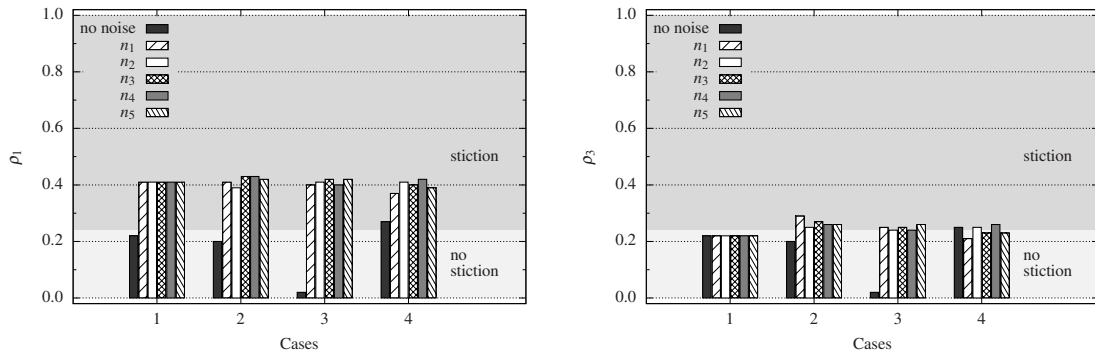
The trends of the indexes  $\rho_1$  and  $\rho_3$  for both Yamashita's method and for the new approach as well as the index  $\gamma$  for the slow dynamics Yamashita's index (Table 3.11) are represented in Figure 3.50 for the cases of no stiction, pure deadband, and stiction with undershoot, and no-offset. Additionally, the indexes obtained for the closed-loop datasets without noise are also illustrated.

In Figure 3.50a (that illustrates the results with Yamashita's method), it is possible to observe that always there is noise in the data (but independently of its

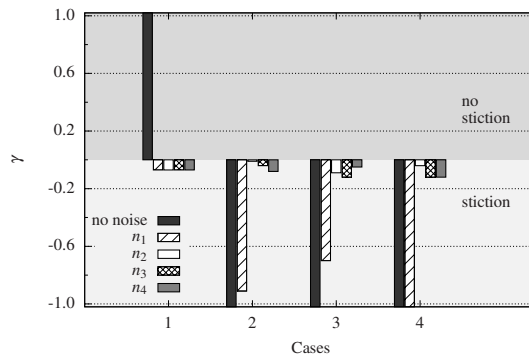
Table 3.11: Influence of noise in stiction detection by the three compared methods. The shadow indicates a wrong detection.

Case	$n_1 \sim \mathcal{N}(0, 0.1^2)$			$n_2 \sim \mathcal{N}(0, 0.2^2)$			$n_3 \sim \mathcal{N}(0, 0.3^2)$			$n_4 \sim \mathcal{N}(0, 0.4^2)$			$n_5 \sim \mathcal{N}(0, 0.5^2)$		
	Indexes	Eval.		Indexes	Eval.		Indexes	Eval.		Indexes	Eval.		Indexes	Eval.	
<i>Yamashita's Method</i>															
No stiction	$\rho_1$	$\rho_3$	$\gamma$	$\rho_1$	$\rho_3$	$\gamma$	$\rho_1$	$\rho_3$	$\gamma$	$\rho_1$	$\rho_3$	$\gamma$	$\rho_1$	$\rho_3$	$\gamma$
	0.41	0.22	✓	0.41	0.22	✓	0.41	0.22	✓	0.41	0.22	✓	0.41	0.22	✓
Pure deadband	0.41	0.29	✓	0.39	0.25	✓	0.43	0.27	✓	0.43	0.26	✓	0.42	0.26	✓
Stiction undershoot	0.40	0.25	✓	0.41	0.24	✓	0.42	0.25	✓	0.40	0.24	✓	0.42	0.26	✓
Stiction no-offset	0.37	0.21	✓	0.41	0.25	✓	0.40	0.23	✓	0.42	0.26	✓	0.39	0.23	✓
<i><math>\gamma</math>. Slower Dynamics</i>															
No stiction		$\gamma$	✓		$\gamma$	✓		$\gamma$	✓		$\gamma$	✓		$\gamma$	✓
		-0.07	✓		-0.07	✓		-0.07	✓		-0.07	✓		-0.07	✓
Pure deadband		-0.91	✓		-0.01	✓		-0.04	✓		-0.08	✓		0.08	×
Stiction undershoot		-0.70	✓		-0.09	✓		-0.12	✓		-0.05	✓		0.15	×
Stiction no-offset		-1.13	✓		-0.04	✓		-0.12	✓		-0.12	✓		-0.04	✓
<i>New Approach</i>															
No stiction	$\rho_1$	$\rho_3$	×	$\rho_1$	$\rho_3$	×	$\rho_1$	$\rho_3$	×	$\rho_1$	$\rho_3$	×	$\rho_1$	$\rho_3$	×
	0.23	0.16	×	0.23	0.16	×	0.23	0.16	×	0.23	0.16	×	0.23	0.16	×
Pure deadband	0.39	0.39	✓	0.42	0.35	✓	0.47	0.43	✓	0.41	0.38	✓	0.39	0.36	✓
Stiction undershoot	0.42	0.38	✓	0.46	0.42	✓	0.44	0.39	✓	0.38	0.31	✓	0.43	0.39	✓
Stiction no-offset	0.58	0.56	✓	0.46	0.41	✓	0.37	0.32	✓	0.37	0.34	✓	0.37	0.31	✓

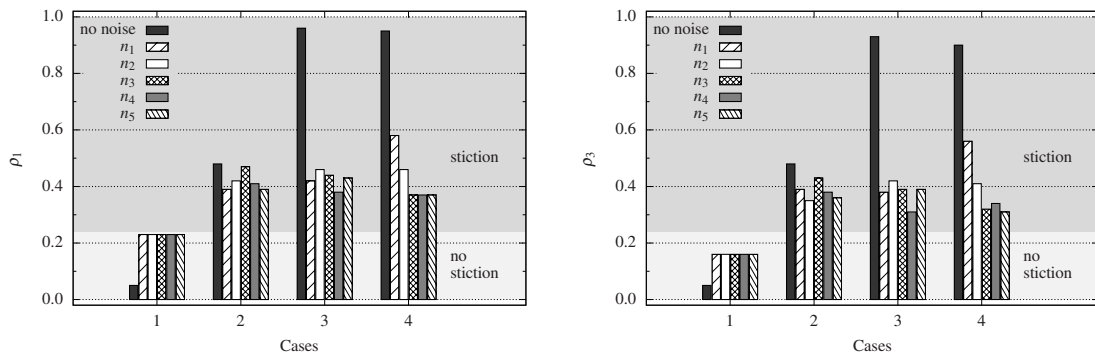




(a) Using Yamashita's method.



(b) Using Yamashita's index for slower dynamics.



(c) Using the proposed approach.

Figure 3.50: Influence of noise for the cases: (1) no stiction, (2) pure deadband, (3) stiction with undershoot, and (4) stiction with no-offset.

intensity) the found indexes are similar even for different cases (no stiction, pure deadband, stiction with undershoot, and stiction with no-offset). In these situations, index  $\rho_1$  is approximately 0.4. This suggests that the method is unable to cope well with the presence of noise reducing the trust on its results. This does not preclude the achievement of correct results, as it was the case of cases 2, 3, and 4 when affected by noise as well as case 1 when the data was not affected by noise. In all the other situations, the method has failed. Moreover, the index  $\rho_3$  is almost always (except case 3, noise free data) around the pre-defined limit of 0.25 and therefore providing very poor information.

Figure 3.50b corresponds to Yamashita's index for slow dynamics and it shows that the method works well when the data is not affected by noise. For noisy data, the method's performance deteriorates rapidly as the index evolves to values close to the pre-defined limit of zero. In some situations, it even turns into wrong conclusions (cases 2 and 3 with the highest intensity of noise). In Figure 3.50c (which corresponds to the new approach proposed), it is possible to observe that the values of the indexes obtained from the no-stiction data are clearly in the no stiction zone ( $0 \leq \rho_i \leq 0.25$ ). The pure deadband renders intermediate values ( $\rho_i \sim 0.5$ ). The case of stiction with undershoot obtains higher values for  $\rho_i$  than the other stiction case, probably justified by the larger jump component in this last case ( $J \geq S$ ). In the presence of noise mitigated with the use of filtering, the indexes maintain correct trends in all the cases, even though an evident influence of the presence/absence of noise may be observed. For instance, for the no-stiction case and in the presence of noise,  $\rho_1$  is very close to 0.25 and almost results in a false positive. Interestingly, for this case of no-stiction, the method seems to be insensitive to the noise intensity, once it is present. In comparison,  $\rho_3$  copes better with the presence of noise and achieves a bigger distance from the limit value for this no-stiction case. In the pure deadband case,  $\rho_i$  values experienced a slight decrease. The most significant change was observed in the stiction cases where the index values were radically reduced to values near the ones obtained by the pure deadband case. Such behavior may be attributed to the fact that the jump component of stiction is hidden by the noise as it has fast dynamics and amplitude compared to the stick component and the process dynamics. Although the present approach is affected by the presence of noise, it showed

Table 3.12: Stiction detection results for level control industrial data by the three compared methods. The shadow indicates a wrong detection of the presence or absence of stiction.

Dataset	True Eval.	Yamashita's Method			Yamashita's index for slower dynamics		New Approach		
		$\rho_1$	$\rho_3$	Eval.	$\gamma$	Eval.	$\rho_1$	$\rho_3$	Eval.
CHEM4	×	0.15	0.09	×	-1.22	✓	0.11	0.00	×
CHEM26	✓	0.03	0.01	×	0.90	×	0.48	0.21	✓
CHEM73	×	0.29	0.15	✓	32.10	×	0.24	0.11	×

adequate performance after a simple data filtering.

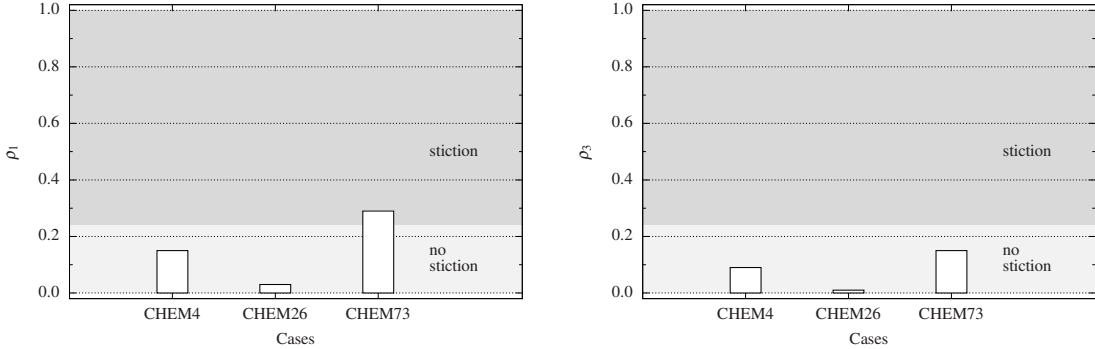
### ■ Using industrial data

The new approach was also applied to three industrial datasets collected by Jelali and Huang (Jelali and Huang, 2013). The first dataset is identified by CHEM4 in Jelali's database and is characterized by containing a controller with tuning problems. The second dataset, identified by CHEM26, corresponds to a control loop containing valve stiction. Finally, the third dataset is identified by CHEM73 and corresponds to a control loop performing well (the root cause of the oscillation is an external disturbance).

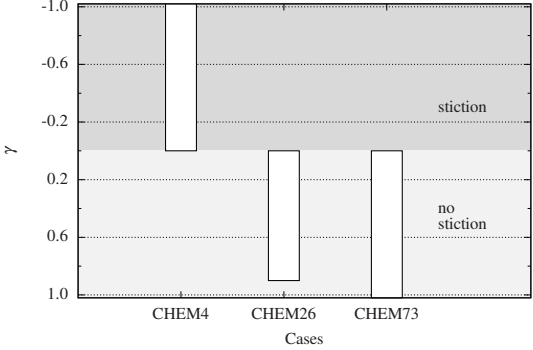
Table 3.12 presents the results obtained by the three methods.

The first case (CHEM4) is correctly undetected by Yamashita's original method, but Yamashita's index for slower dynamics produces a false positive. As for the case CHEM26, both methods fail in detecting the existence of stiction. In what concerns the case CHEM73, the first method fails while the second indicates a correct negative result. These results show that these two methods don't consistently detect the presence/absence of stiction. However, the new approach was able to diagnose correctly all the cases under consideration.

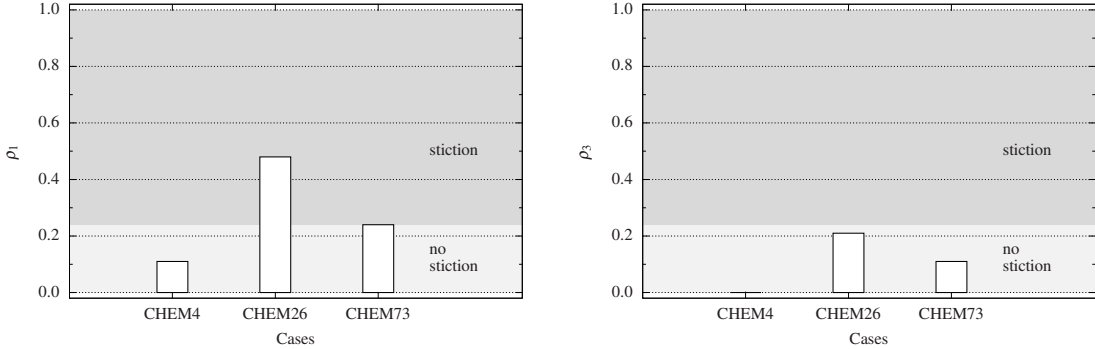
Figure 3.51 shows graphically these results. Although the new approach detects all cases correctly, it is noteworthy that the index  $\rho_1$  is very close to the pre-defined limit for case CHEM73. In opposition, index  $\rho_3$  provides an unequivocal conclusion of absence of stiction for that case. Figure 3.51 also makes it clear that Yamashita's method identifies incorrectly CHEM26 and CHEM73 cases and Yamashita's index fails to identify CHEM26 case.



(a) Using the Yamashita's method.



(b) Using Yamashita's index for slower dynamics.



(c) Using the proposed approach.

Figure 3.51: Stiction detection results for industrial data: CHEM4 (no stiction), CHEM26 (stiction), and CHEM73 (no stiction).

# Chapter 4

## PID controllers tuning

The diagnosis of problems in final control elements (addressed in Chapter 2) plays an extremely important role in maintaining high performance of control loops. However, such efforts are completely useless if, once the fault is detected and characterized, a compensation or remedy is not activated. In this context, this Chapter is concerned with the development of a tuning method of Proportional-Integral-Derivative (PID) controllers that considers explicitly the fault potentially existing in the final control element and provides a compensation strategy while valve maintenance is not possible.

Beforehand, a number of PID algorithm formulations are introduced together with some particular aspects of their digital computer implementation. Classic tuning methods and their extensions, covering single and multiple control loops, are discussed.

### 4.1 The PID controller

The PID controller is unquestionably the most common controller algorithm used in industry. In fact, more than 95% of the controllers used in process industries are PID algorithms or its enhanced versions (Yamamoto and Hashimoto, 1991; Åström and Hägglund, 2004; Eriksson and Koivo, 2005). Its predominance relies in three main reasons. PID controllers work very well in most of the systems. They are simple to understand and implement. Moreover, these algorithms are pre-programmed in every control systems (Desborough and Miller, 2002). In

spite of its widespread use, there exists no generally accepted design method for the controller. However, its design is a crucial point because a suboptimal structure and tuning may result not only in undermined control performance but also may originate process instability. The PID controller combines three types of control actions: proportional, integral, and derivative. The proportional action is expressed as

$$u = k_C e, \quad (4.1)$$

where  $k_C$  is the proportional gain, has the advantage of providing small control efforts comparatively to on-off control, but it also produces a steady-state error. Commercial products may use proportional band defined as  $PB = 100/k_C$  instead of the proportional gain constant. The integral action is proportional to the integral of the control error

$$u = \frac{1}{\tau_I} \int_0^t e \, dt, \quad (4.2)$$

where  $\tau_I$  is the integral time. It relates to past values of the control error and eliminates the steady-state error generated by the proportional action. For this reason, this action is also called automatic reset. The derivative action is expressed as

$$u = \tau_D \frac{de}{dt}, \quad (4.3)$$

where  $\tau_D$  is the derivative time, and may improve the control performance anticipating the undesired trend of the control error. This action may also be called as anticipatory control or rate action. The combination of the three actions in various configurations results in the PID controller structure.

According to Tan (1999), one important reason for the non-standard structures is the transition of the controllers from pneumatic implementation through electronic implementation to the present microprocessor implementation. Figure 4.1 shows a simplified block diagram of the PID controller inside a control loop. The control law as structured in Figure 4.1 is defined by the sum of the three components as

$$u = k_C e + \frac{1}{\tau_I} \int_0^t e \, dt + \tau_D \frac{de}{dt}, \quad (4.4)$$

which is commonly called as the parallel form of the ideal PID controller and may

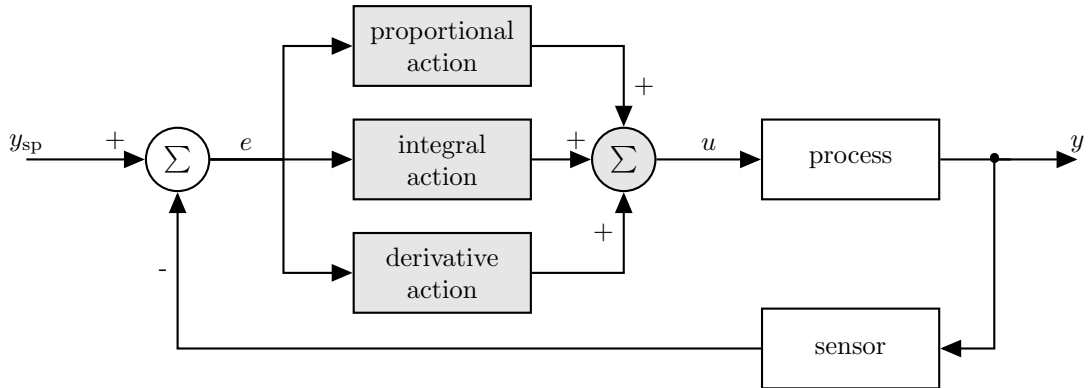


Figure 4.1: Block diagram highlighting a PID controller inside the feedback control loop.

be rewritten by the application of the Laplace transformation  $\mathcal{L}(\cdot)$  as

$$G_C(s) = \frac{U(s)}{E(s)} = k_C + \frac{1}{\tau_I s} + \tau_D s, \quad (4.5)$$

where  $s$  is the Laplace frequency variable,  $U(s) = \mathcal{L}(u)$ , and  $E(s) = \mathcal{L}(e)$ .

Other structures commonly used are the series form of the ideal PID controller described as

$$G_C(s) = k_C \left( 1 + \frac{1}{\tau_I s} + \tau_D s \right), \quad (4.6)$$

and the series form of the ideal PID controller with a first-order lag given by

$$G_C(s) = k_C \left( 1 + \frac{1}{\tau_I s} + \tau_D s \right) \frac{1}{\tau_f s + 1}. \quad (4.7)$$

A substantial number of the variations are detailed in O'Dwyer (2006) identifying some software where those variations are implemented.

Usually, the proper combination (P, PI, PD, PID) of the three actions is part of the controller design that considers process specifics and the control objectives. For instance, the P controller is simple to design and may be the best choice for some applications where the steady-state error is not a concern, such as in surge tank levels. But if the zero steady-state error is a requirement, the PI controller provides a sufficiently good performance for the most of processes. The most complete combination, PID controller, may provide a significant improvement

of the performance in processes with deadtime and slow dynamics. However, derivative action has associated some problems related to the measurement noise filtering and a more complicated tuning procedure.

Modern control systems implement digital versions of the control law derived essentially from the approximations

$$\int_0^t e \, dt \approx \sum_{i=1}^k e_i \Delta t, \quad (4.8)$$

$$\frac{de}{dt} \approx \frac{e_k - e_{k-1}}{\Delta t}, \quad (4.9)$$

where  $\Delta t$  is the sampling period and  $e_k$  is the control error at the  $k^{\text{th}}$  iteration. These approximations are used to generate two alternative forms of the digital PID control law: the position and the velocity form given by

$$u_k = \bar{u} + k_C \left[ e_k + \frac{\Delta t}{\tau_I} \sum_{i=1}^k e_i + \frac{\tau_D}{\Delta t} (e_k - e_{k-1}) \right], \quad (4.10)$$

and

$$u_k = u_{k-1} + k_C \left[ (e_k - e_{k-1}) + \frac{\Delta t}{\tau_I} e_k + \frac{\tau_D}{\Delta t} (e_k - 2e_{k-1} + e_{k-2}) \right], \quad (4.11)$$

respectively, where  $\bar{u}$  is the steady-state manipulated variable. Commonly, the velocity form is preferable because it inherently contains anti-reset windup and does not require the initialization of  $\bar{u}$  (Seborg et al., 2010).

The above velocity formulation is prone to the so called “proportional and derivative kicks”(Johnson and Moradi, 2006; King, 2011), that is, to a spike in the manipulated variable when a step change is introduced in the setpoint. This problem may be handled by substituting the control error in (4.11) with the controlled variable in the respective control law terms. This gives rise to formulations that clearly separate the terms that are based on the control error from those based on the controlled variable, reflecting this in the naming: [terms based on control error]-[terms based on controlled variable]. The derivative kick may be removed



by using a PI-D controller defined by

$$u_k = u_{k-1} + k_C \left[ (e_k - e_{k-1}) + \frac{\Delta t}{\tau_I} e_k + \frac{\tau_D}{\Delta t} (y_k - 2y_{k-1} + y_{k-2}) \right], \quad (4.12)$$

where the proportional and integral actions are based on the control error and the derivative action is based on the controlled variable. King (2011) refers to this last approach as the derivative-on-pv version and explains that this improvement is usually standard and must be preferred instead of the derivative action based on the error. The proportional kick may be removed adding the proportional term based on the controlled variable. The resulting I-PD controller

$$u_k = u_{k-1} + k_C \left[ (y_k - y_{k-1}) + \frac{\Delta t}{\tau_I} e_k + \frac{\tau_D}{\Delta t} (y_k - 2y_{k-1} + y_{k-2}) \right], \quad (4.13)$$

will not produce a sharp change because the proportional term is not affected by the control error. King (2011) calls this equation as the proportional-on-pv version of the PID control law and states that it is the most misunderstood and most underutilised version.

## 4.2 State-of-the-art

The present section aims to review some of the most used methods for the PID controller tuning.

### 4.2.1 Tuning methods

It is well known that good PID controller performance is necessary in order to ensure safety and compliance as well as to achieve the economic objectives of production lines. Therefore, it is essential to monitor PID controller performance and, when necessary, carry out corrective actions, such as its retuning.

A considerable number of PID controller tuning methods is available for the design of process control systems. They should be used instead of the common trial-and-error approach that is time consuming and that does not achieve the optimal performance, especially of the inherently multivariate control systems (Dittmar et al., 2012). However, the practitioner should take care because

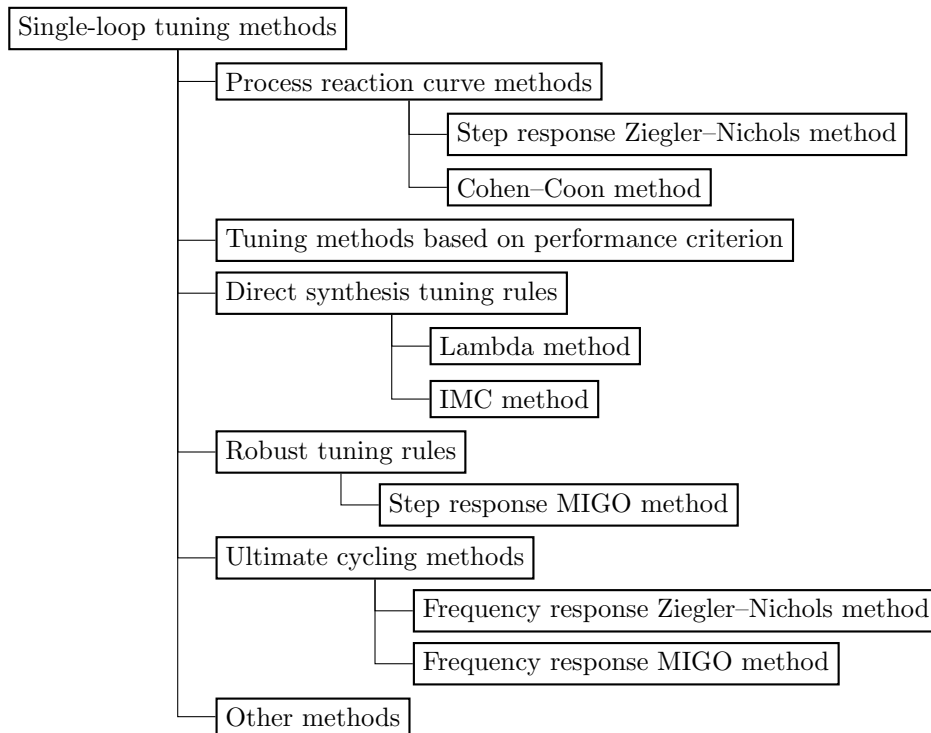


Figure 4.2: Classification of single-loop tuning methods with some examples of existing methods.

many tuning rules assume that the PID controller equations are that of the ideal PID controller structure. But, as a matter of fact, there is substantial variation among the vendors, as O’Dwyer (2006) explains.

For a design method to be efficient, it has to be applicable to a wide range of systems and must have the capability of considering specific control problem specifications. Therefore, it must be robust either by providing the controller parameters in case they exist or by informing that the specifications may not be met (Åström et al., 1998).

Some PID controller tuning methods are explored below.

### ■ Single-loop tuning methods

O’Dwyer (2006) classified the tuning rules for SISO processes in six main divisions: process reaction curve methods, tuning methods based on minimizing an appropriate performance criterion, direct synthesis tuning rules, robust tuning rules and ultimate cycling methods. Figure 4.2 depicts this classification.

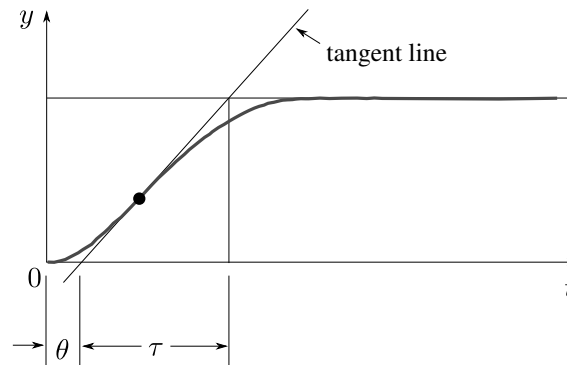


Figure 4.3: Measurement of variables  $\theta$  and  $\tau$  from the system step response for step response Ziegler-Nichols method.

Process reaction curve tuning rules are based on the calculation of controller parameters from the model parameters determined from the open-loop process step response. The advantages of such tuning strategies are that only a single experimental test is necessary, a trial-and-error procedure is not required and the controller settings are easily calculated. However, it is not trivial to determine an accurate and parsimonious process model and to account for load changes that may occur during the test, distorting the test results. Besides, a large step input may be necessary to achieve a good signal to noise ratio (Hang et al., 1991; O'Dwyer, 2003).

The well known Ziegler–Nichols and Cohen–Coon methods are examples of process reaction methods.

Ziegler–Nichols methods were originally suggested by Ziegler and Nichols (1942) and are still widely used in the process industry as the basis for the controller tuning (Eriksson, 2008). The classical methods of Ziegler–Nichols are the step and frequency response methods. Based on a SISO process described by a first-order plus time delay (FODT) model, Ziegler and Nichols (1942) developed the **step response Ziegler–Nichols method** (method that fits in the first class of the classification in Figure 4.2) simulating a large number of different processes and correlating the controller parameters with the features of the step response as reviewed by Åström and Hägglund (2004). The method based on an open-loop step response test of the process requires that the process be stable. The step response (Figure 4.3) is characterized by two process parameters determined by

Table 4.1: Step response Ziegler–Nichols tuning method formulae for self-regulating and integrating processes, where  $\theta$  is the process time delay and  $\tau$  is the process time constant.

Type	$k_C$	$\tau_I$	$\tau_D$
P	$\theta/\tau$	$+\infty$	0
PI	$0.9\theta/\tau$	$\tau/0.3$	0
PID	$1.2\theta/\tau$	$2\tau$	$0.5\tau$

Table 4.2: Cohen–Coon method formulae for self-regulating and integrating processes, where  $K_p$  is the process static gain.

Type	$k_C$	$\tau_I$	$\tau_D$
P	$\frac{1}{K_p} \frac{\tau}{\theta} \left[ 1.0 + \frac{\theta}{3\tau} \right]$	$+\infty$	0
PI	$\frac{1}{K_p} \frac{\tau}{\theta} \left[ 0.9 + \frac{\theta}{12\tau} \right]$	$\theta \left[ \frac{30 + 3\frac{\theta}{\tau}}{9 + 20\frac{\theta}{\tau}} \right]$	0
PID	$\frac{1}{K_p} \frac{\tau}{\theta} \left[ \frac{4}{3} + \frac{\theta}{4\tau} \right]$	$\theta \left[ \frac{32 + 6\frac{\theta}{\tau}}{13 + 8\frac{\theta}{\tau}} \right]$	$\theta \left[ \frac{4}{11 + 2\frac{\theta}{\tau}} \right]$

drawing a tangent line at the inflexion point, where the slope of the step response has its maximum value. The intersection of the abscissae axis and the tangent is used to calculate the controller parameters (Eriksson, 2008) according to the expressions summarized in Table 4.1.

The **Cohen–Coon method** developed by Cohen and Coon (1953), based on the FODT process model and analytical and numerical computations, considers the rejection of load disturbance as the main objective and requires that the process be stable. It minimizes the integral of error due to a unit step load disturbance subject to the restriction of a decay ratio of a quarter amplitude. The corresponding expressions are given in Table 4.2.

Tuning rules based on minimizing a certain performance criterion may be defined to optimize the regulatory response, servo response or other characteristics of a compensated delayed process. The minimization of the integral of absolute error (IAE), the integral of squared error (ISE), and the integral over time of the

absolute error (ITAE) in a closed loop control are examples of such criteria. Lopez et al. (1967) and Smith (1972) developed the **penalty functions based method** for improving control loops performance based on these three criteria and in the presence of load or setpoint disturbances. In the presence of load disturbances, the rules

$$k_C = \frac{A \left( \frac{\theta}{\tau} \right)^B}{K_p}, \quad (4.14)$$

$$\tau_I = \frac{\tau}{A \left( \frac{\theta}{\tau} \right)^B}, \quad (4.15)$$

$$\tau_D = A \left( \frac{\theta}{\tau} \right)^B \tau, \quad (4.16)$$

are used for the tuning of P, PI, and PID controllers parametrized by constants in Table 4.3. In opposition, rules (4.14), (4.16), and

$$\tau_I = \frac{\tau}{A + B \left( \frac{\theta}{\tau} \right)} \quad (4.17)$$

are applied when the closed loop process is subject to setpoint disturbances (Table 4.3).

Direct synthesis tuning rules give a desired closed loop response specifying a time domain related metric, such as the desired poles of the closed loop response, or a frequency domain metric, such as a specified gain margin and/or phase margin. Lambda and IMC methods belong to the class of direct synthesis tuning rules.

The **Lambda method**, originally proposed by Dahlin (1968) as a special case of pole placement, is widespread in the process industry because it is simple and may explicitly specify the closed loop response time, commonly chosen as three times the open-loop process time constant. The process identification is done with a FOTD model approximating the delay with the Taylor series expansion or the Padé approximation. The main drawback of this rule is the poor response to load disturbances for lag dominated systems due to the process pole cancellation. The Lambda method formulae are presented in the first line of Table 4.4.

Table 4.3: Performance criteria based tuning method constants.

Type	$k_C$		$\tau_I$		$\tau_D$	
	A	B	A	B	A	B
<i>IAE for load disturbances</i>						
P	0.902	-0.985	–	–	–	–
PI	0.984	-0.986	0.608	-0.707	–	–
PID	1.435	-0.921	0.878	-0.749	0.482	1.137
<i>ISE for load disturbances</i>						
P	1.411	-0.917	–	–	–	–
PI	1.305	-0.959	0.492	-0.739	–	–
PID	1.495	-0.945	1.101	-0.771	0.560	1.006
<i>ITAE for load disturbances</i>						
P	0.490	-1.084	–	–	–	–
PI	0.859	-0.977	0.674	-0.680	–	–
PID	1.357	-0.947	0.842	-0.738	0.381	0.995
<i>IAE for setpoint disturbances</i>						
PI	0.758	-0.861	1.020	-0.323	–	–
PID	1.086	-0.869	0.740	-0.130	0.348	0.914
<i>ITAE for setpoint disturbances</i>						
PI	0.586	-0.916	1.030	-0.165	–	–
PID	0.965	-0.855	0.796	-0.147	0.308	0.929

The **Internal Model Control (IMC) method** is a model-based control method. The method name derives from the fact that the controller contains a process model internally. A process is controlled with IMC controller introducing a filter to obtain a closed-loop system less sensitive to modeling errors (Lee et al., 1998). This method considers robustness explicitly through the use of a proper filter (Åström and Hägglund, 2006). Since the method implies that the poles and zeros are canceled, the system response to load disturbances may be poor in load disturbances. The corresponding formula are described in Table 4.4.

All tuning methods discussed so far have in common the need to check the robustness to process variations after the design. On the contrary, robust tuning rules have an explicit robust stability and/or robust performance criterion built into the design process.

The **step response MIGO (M constrained integral gain optimization) method**, proposed by Åström and Hägglund (2004), maximizes the integral gain subject to robustness constraints. This design method is suitable for systems where the major concern is the load disturbance rejection. Although setpoint changes or

Table 4.4: IMC tuning method formulae for self-regulating and integrating processes, where  $\lambda$  is the desired process time constant response.

Type	Self-regulating process			Integrating process		
	$k_C$	$\tau_I$	$\tau_D$	$k_C$	$\tau_I$	$\tau_D$
PI	$\frac{1}{K_p} \frac{\tau}{\lambda + \theta}$	$\tau$	0	$\frac{1}{K_p} \frac{2\lambda + \theta}{(\lambda + \theta)^2}$	$2\lambda + \theta$	0
PID (non-interactive)	$\frac{1}{K_p} \frac{\tau + \frac{\theta}{2}}{\lambda + \theta}$	$\tau + \frac{\theta}{2}$	$\frac{\tau\theta}{2\tau + \theta}$	$\frac{1}{K_p} \frac{2\lambda + \theta}{\left(\lambda + \frac{\theta}{2}\right)^2}$	$2\lambda + \theta$	$\frac{\lambda\theta + \frac{\theta^2}{4}}{2\lambda + \theta}$
PID (interactive)	$\frac{1}{K_p} \frac{\tau}{\lambda + \theta}$	$\tau$	$\frac{\theta}{2}$	$\frac{1}{K_p} \frac{2\lambda + \frac{\theta}{2}}{\left(\lambda + \frac{\theta}{2}\right)^2}$	$2\lambda + \frac{\theta}{2}$	$\frac{\theta}{2}$

Table 4.5: Step response MIGO tuning method formulae for self-regulating and integrating processes.

Type	Self-regulating process			Integrating process		
	$k_C$	$\tau_I$	$\tau_D$	$k_C$	$\tau_I$	$\tau_D$
PI	$\frac{0.15}{K_p} + \left(0.35 - \frac{\theta\tau}{(\theta + \tau)^2}\right) \frac{\tau}{K_p\theta}$	$0.35\theta + \frac{13\theta\tau^2}{\tau^2 + 12\theta\tau + 7\theta^2}$	0	$\frac{0.35\tau}{K_p\theta}$	13.4 $\theta$	0
PID	$\frac{1}{K_p} \left(0.2 + 0.45\frac{\tau}{\theta}\right)$	$\frac{0.4\theta + 0.8\tau}{\theta + 0.1\tau}\theta$	$\frac{0.5\theta\tau}{0.3\theta + \tau}$	$\frac{0.45\tau}{K_p\theta}$	8 $\theta$	0.5 $\theta$

noise are not taken into account in the method development, the authors presented guidelines to handle these aspects. The developed formulae for the step response MIGO method are shown in Table 4.5.

Finally, the ultimate cycle tuning rules are based on recording appropriate parameters at the ultimate frequency (that is, the frequency at which marginal stability of the closed loop control system occurs). The first tuning rule was defined in Ziegler and Nichols (1942). The frequency response Ziegler–Nichols method describes the process with two parameters: the ultimate gain,  $k_{C,u}$ , and the ultimate period,  $T_u$ . To determine these parameters, the plant is controlled with a P controller whose gain is increased until the system oscillates critically. The ultimate period is the period of oscillation at the ultimate gain. The ultimate gain is the proportional gain that yields the marginal stability. The need to bring the

Table 4.6: Frequency response Ziegler–Nichols tuning method formulae for self-regulating and integrating processes.

Type	$k_C$	$\tau_I$	$\tau_D$
P	$0.50k_{C,u}$	$+\infty$	0
PI	$0.45k_{C,u}$	$T_u/1.2$	0
PID	$0.60k_{C,u}$	$T_u/2.0$	$T_u/8.0$

process to a sustained oscillation is a critical drawback of the method because it may be potentially harmful. Fortunately, it is possible to determine the ultimate constants from the process dynamics using the following relationships:

Self-regulating processes:

$$k_{C,u} = -\frac{1}{K_p \cdot \cos\left(\frac{2\pi\theta}{T_u}\right)}, \quad (4.18)$$

$$\frac{2\pi\tau}{T_u} + \tan(2\pi\theta T_u) = 0, \quad (4.19)$$

Integrating processes:

$$k_{C,u} = \frac{\pi}{2K_p\theta}, \quad (4.20)$$

$$T_u = 4\theta. \quad (4.21)$$

The determination of  $T_u$  requires an iterative process (see Section 3.10 of King (2011) for practical issues). Finally, the tuning parameters are derived from simple calculations as shown in Table 4.6.

The MIGO method determines a set of rules for the frequency response of processes whose gain ratio  $\kappa = 1/(K_p k_{C,u})$  is greater than 0.4. The corresponding formulae are given in Table 4.7.

More advanced performance criteria were considered in other methods available in the literature. For instance, Zareba et al. (2014) proposed an intuitive technique to tune PID controllers based on the Harris index to improve the controller performance in a realistic process/simulation environment. The controller behaviour is characterized automatically by calculating the relative damping index



Table 4.7: Frequency response MIGO tuning method formulae for self-regulating and integrating processes.

Type	$k_C$	$\tau_I$	$\tau_D$
PI	$0.16k_{C,u}$	$\frac{1}{1 + 4.5\kappa}T_u$	0
PID	$(0.3 - 0.1\kappa^4)k_{C,u}$	$\frac{0.6}{1 + 2\kappa}T_u$	$\frac{0.15(1 - \kappa)}{1 - 0.95\kappa}T_u$

defined as

$$\text{RDI} = \frac{D_{\text{IR}} - D_{\text{IR,agg}}}{D_{\text{IR,slug}} - D_{\text{IR}}}, \quad (4.22)$$

where  $D_{\text{IR}}$  is the damping factor of the system impulse response,  $D_{\text{IR,agg}}$  and  $D_{\text{IR,slug}}$  are user defined damping factor limits for aggressive and sluggish controller behaviour, respectively, specifying the controller performance region (usually assuming the following values: 0.6 and 0.8 for self-regulating processes, and 0.3 and 0.5 for integrating processes). The interpretation of the damping factor index is performed as: the control performance is good when  $\text{RDI} > 0$ , the control is aggressive when  $-1 \leq \text{RDI} \leq 0$ , and the control is sluggish when  $\text{RDI} < -1$ .

The method is presented in the flowchart of Figure 4.4. Here, the impulse response is obtained by fitting a time series model of type AR or ARMA to the measured closed-loop output data (see Chapter 2). Then, a second fitting is performed to the time series model coefficients using the second-order model considering time delay given by

$$G(s) = \frac{K_p e^{-\theta s}}{\tau^2 s^2 + \tau D_{\text{IR}} s + 1}, \quad (4.23)$$

where  $\theta$  is the time delay,  $K_p$  is the static gain, and  $\tau$  is the time constant. The value of  $D_{\text{IR}}$  necessary in (4.22) is inferred from the values of (4.23) coefficients. PID controller parameters may be varied according to different strategies: variation of proportional gain alone and fine tuning of the integral time, simultaneous variation, and successive variation. The strategies used by the method are defined by the user.

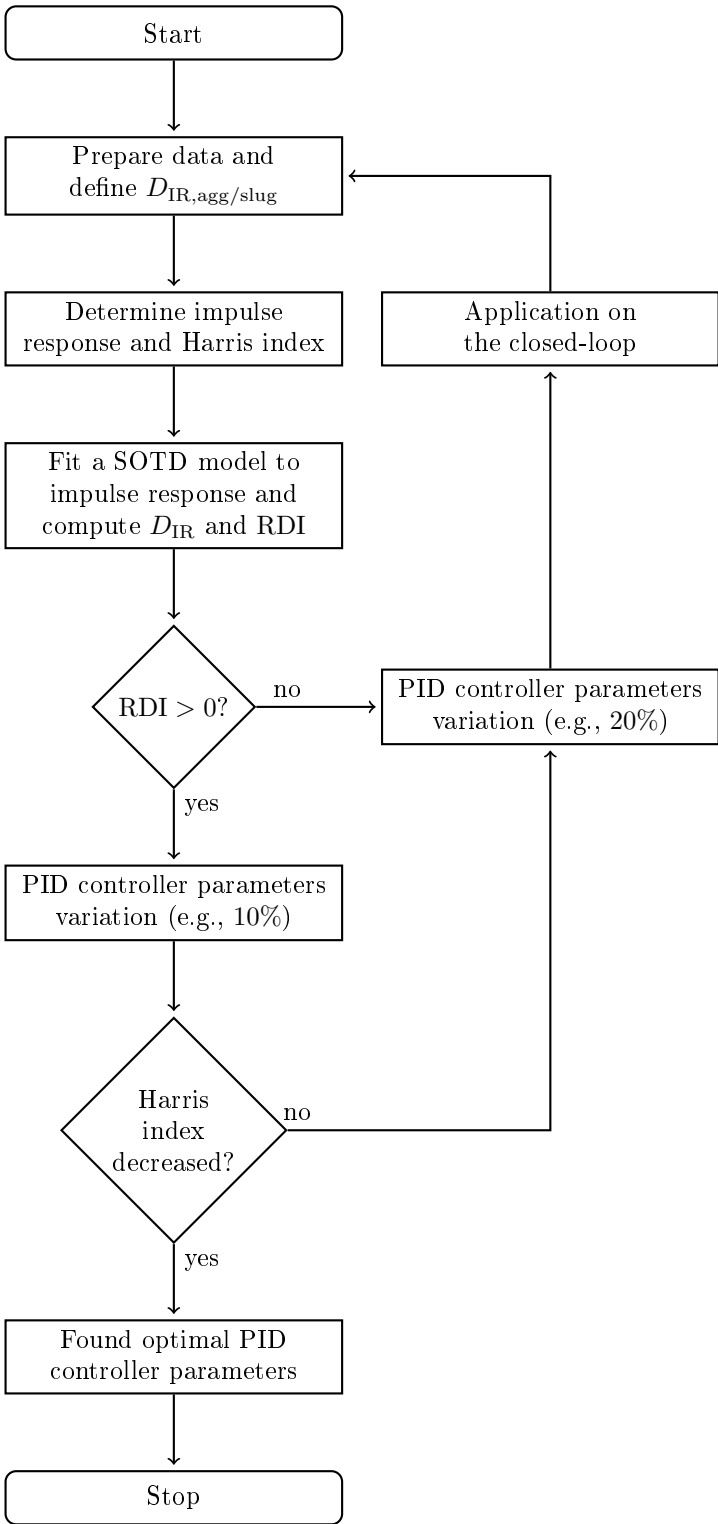


Figure 4.4: Zareba method flowchart (Zareba et al., 2014).

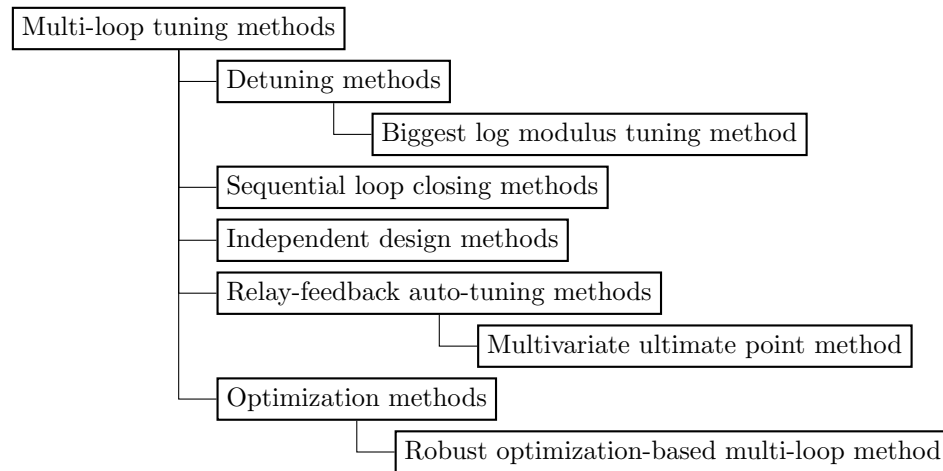


Figure 4.5: Classification of multi-loop tuning methods with some examples of existing methods.

### ■ Multi-loop tuning methods

The tuning of PID controllers considering the system as SISO in a multivariate environment is usually done in a time-consuming, sequential and iterative way. Interactions among the process variables may make this task difficult and the results strongly depend on the peculiarities of the application and on the experience of the engineer. Dittmar et al. (2012) presents a review of design methods that have been developed for multi-loop tuning. According to these authors, they may be classified into: detuning, sequential loop closing, independent design, relay-feedback auto-tuning, and optimization methods (Figure 4.5).

The detuning methods consist of the design of each PID controller based on its process transfer function and ignoring interactions from another loops. The controllers are then detuned to take into account these interaction.

The most popular method is called **biggest log modulus tuning method** and was developed by Luyben (1986). In order to determine PID controller constants that generate good setpoint and load responses in the multivariable system, the method ensures a sufficient margin of stability to the control loops by maximizing the distance between the system and the instability point in a Nyquist plot. To do that, the Ziegler–Nichols PID controller parameters are detuned to maximize the

multivariable closed-loop log modulus defined as

$$L_{cm} = 20 \log \left| \frac{w}{1+w} \right|, \quad (4.24)$$

guaranteeing its maximum value of  $2n$ , where  $n$  denotes the dimension of the multivariable system and  $w$  is a function given by

$$w = -1 + \det(I + GC), \quad (4.25)$$

where  $I$  is the identity matrix, and  $G$  and  $C$  are the process and controller transfer matrices, respectively. The method review in Section 9.2.1 of Schork (1993) explains in detail the algorithm.

Hovd and Skogestad (1994) define the sequential loop closing methods as methods that close the loops sequentially starting with the fastest loop. The disadvantage of these methods is that the results are dependent on the order of the loop closing and on the method used for each controller design.

Using independent design methods, both Hovd and Skogestad (1993) and Chen and Seborg (2003) considered loop interactions, robust performance and stability using a first interaction analysis and then designed the controllers one by one.

Relay-feedback automatic tuning methods are applicable if no analytical process model exists. However, the necessity of sequential or simultaneous relay-feedback experiments under industrial conditions limit them. Campestrini et al. (2009) and Palmor et al. (1995) works are examples of these methods.

The **multivariate ultimate point method** provided by Campestrini et al. (2009) extended the ultimate quantities based methods to the multivariable case. Although other approaches had already performed this extension, they were not consistent because they generate several ultimate points for the MIMO system that are applied to the SISO formula or they develop the MIMO system tuning based on one of the determined ultimate points resulting in their closed-loop performance strongly related to that particular point. Consider the MIMO square processes described by

$$Y(s) = G(s)U(s), \quad (4.26)$$

where  $U(\cdot)$  and  $Y(\cdot)$  are the Laplace transforms of the process input and output variables, respectively, and  $G(\cdot)$  is the process transfer matrix calculated from data generated by decentralized relay feedback experiments. The first step to obtain PID controller parameters is to solve

$$a_{m-i}\Lambda^i = \sum M_i \cdot G(jw_u) \cdot C(jw_u), \quad (4.27)$$

in order to the controller transfer matrix  $C(jw_u)$ , where  $m$  is the order process,  $w_u$  is the ultimate frequency (related to the ultimate period by  $T_u = 2\pi/w_u$ ),  $j$  is the imaginary unit equal to  $\sqrt{-1}$ ,  $M_i$  are the  $i^{\text{th}}$  principal minors of

$$G(jw_u)C(jw_u) = \begin{bmatrix} g_{11}(jw_u) & \cdots & g_{1m}(jw_u) \\ g_{21}(jw_u) & \cdots & g_{2m}(jw_u) \\ \vdots & \ddots & \vdots \\ g_{m1}(jw_u) & \cdots & g_{mm}(jw_u) \end{bmatrix} \cdot \begin{bmatrix} p_1(jw_u) & \cdots & 0 \\ 0 & p_2(jw_u) & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & p_m(jw_u) \end{bmatrix}, \quad (4.28)$$

(each sum is taken over all principal minors of order  $i$ ), and  $a_{m-1}$  are the coefficients of the characteristic equation  $(s - \lambda)^m$  obtained using the Newton binomial formula  $a_{m-i} = \frac{m!}{(m-i)!i!}$ . From (4.27) and (4.28), its possible to determine  $p(jw_u)$ . Since  $p(jw_u)$  is defined by

$$p_k(jw_u) = k_{C,k} \left[ 1 + \left( \tau_{D,k}w_u - \frac{1}{\tau_{I,k}w_u} \right) j \right], \quad k = 1, \dots, m, \quad (4.29)$$

the PID controller parameters are now calculated by

$$k_{C,k} = \text{Re}\{p_k(jw_u)\}, \quad (4.30)$$

$$k_{C,k} \left( \tau_{D,k}w_u - \frac{1}{\tau_{I,k}w_u} \right) = \text{Im}\{p_k(jw_u)\}. \quad (4.31)$$

Considering the usual rule in the tuning of SISO systems  $\tau_D = \tau_I/4$ ,  $\tau_D$  and  $\tau_I$  may now be determined and the complete tuning is achieved.

Optimization methods are recent developments that consider an analytical process model and a controller structure and close the loop via the minimization a pre-defined objective function. However, a solver is required to find the solution of the optimization problem. Most common solvers are based on numerical

methods, such as least squares or sequential quadratic programming (Dittmar et al., 2012). Also, evolutionary algorithms based on mechanisms inspired by biological evolution are used by various authors (Sumana and Venkateswarlu, 2010). Dittmar et al. (2012) classify these particular algorithms by their performance as global search methods. The use of optimization methods brings advantages as a less conservative controller design and the overall nominal stability naturally achieved.

The **robust optimization-based multi-loop method** described by Dittmar et al. (2012) identifies the full dynamic model of the multivariable system and uses constrained nonlinear optimization techniques to find the controller parameters. The PID controller parameters are calculated by solving the optimization problem

$$\begin{aligned}
 & \underset{k_{C,k}, \tau_{I,k}, \tau_{D,k}}{\text{minimize}} && J && (4.32) \\
 & \text{subject to} && && \\
 & && g_j(k_{C,k}, \tau_{I,k}, \tau_{D,k}) \leq 0 &&
 \end{aligned}$$

where  $J$  is the objective function defined by the weighted sum of terms  $J_1$ ,  $J_2$  and  $J_3$  referring to the IAE for setpoint tracking, the IAE for the input step disturbance, and the control effort, respectively. Weighting factors are 1,  $\alpha$  and  $\beta$ , respectively. Nonlinear inequality constraints  $g_j$  may be carefully selected from the following list: maximum controller output deviation after setpoint changes, maximum overshoot on the process variable after setpoint changes, minimal damping or maximum decay ratio, maximal measurement noise amplification, combined process gain and deadtime safety margins, and maximum/minimum limits of the controller parameters. These possible constraints allow the user to meet the intended requirements. The initialization of the controller parameters may be done by the user or obtained by the Cohen-Coon method. If the starting iteration is infeasible, the initialization is done by one of two global search approaches: grid search and genetic algorithms. The optimization problem is solved using a gradient-free direct search method similar to the Nelder–Mead Simplex algorithm.

### 4.2.2 Automatic tuning

Tuning-on-demand is a tedious task because the controller must be retuned periodically as well as whenever changes are introduced in the process. Additionally, under-performance may be detected too late (Li et al., 2006).

These disadvantages may be solved using PID automatic tuning that is a technology with the benefits of structural and implementation simplicity. However, it has not been widely applied in industrial practice. Some of the hindrance to its acceptance is the conservative paradigm *one size fits all* (Bobál, 2005). Besides, the applicability of automatic tuning may be limited in processes with strong changes in operating conditions or lacking of information rich process data. Besides, automatic tuning requires a carefully supervised start-up and testing period. Nevertheless, once the controller is correctly set up, the tuner may constantly monitor the process and automatically adjust the PID controller parameters.

The most widely used automatic tuning methods are based on process reaction curve and on rules because they yield the fastest tuning (Li et al., 2006).

## 4.3 Compensation of control valve faults by PID controller tuning

Control loop performance may be affected by faults in control valves. Stiction is the most common and one of the long-standing faults in the process industry causing persistent oscillations and undermining the control loops performance (Brásio et al., 2014). Of course, the best solution for faults in control valves is to perform maintenance work on the equipment (Gerry, 2002). However, it may be impractical to take a faulty valve out of service until the next turnaround and, therefore, fault accommodation approaches are very desirable.

The effect of stiction may be mitigated with the use of specially crafted signals added to the manipulated variable, the adjustment of the manipulated variable (Srinivasan and Rengaswamy, 2010), or the addition of a special block to the nominal PID algorithm (Farenzena and Trierweiler, 2010). However, this adaptation is not known to the nominal controller and, therefore, may affect the control loop performance causing instability and/or additional wear of mechanical parts.

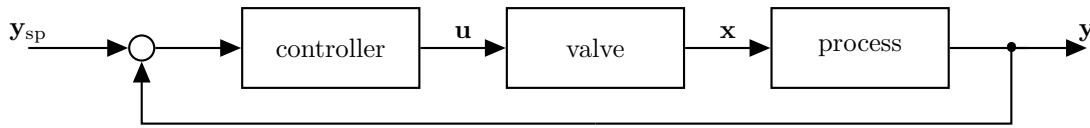


Figure 4.6: Industrial control loop.

Another way of dealing with stiction is the manual detuning of the PID controller in order to reduce the oscillation effect (Mohammad and Huang, 2012). However, while the oscillations effect may decrease (amplitude and frequency reduction), the closed loop performance of the process may also become worse. Besides, manual PID tuning is a time consuming task and, unfortunately, this is common practice in the majority of the plants.

The present section describes an automated method that allows to optimize the control loop performance in the presence of control valve faults. It retunes the controller via numerical constrained optimization.

### 4.3.1 Proposed approach

Usually, an industrial control loop integrates a PID controller, a final control element (control valve) and an industrial process (as depicted in Figure 4.6). Based on the value of the setpoint  $y_{sp}$ , on the integral of the past errors, and on the derivative of the error or the process variable, the PID controller calculates a control action (manipulated variable,  $u$ ) to be executed by the control valve resulting in a process response (controlled variable,  $y$ ). If the control valve is not in perfect conditions, the controller order is not executed exactly by the valve and the real position,  $x$ , does not match  $u$ .

In opposition to the previously published optimization based methods (see Section 3.2.4), the approach presented below determines a set of tuning parameters of the controller taking explicitly into account not only the process dynamics and the control law itself, but also the existence and the magnitude of the faults in the final control element. The tuning results in the best achievable performance of the controller even if the control valve is affected by stiction. It is noteworthy that no structural change of the control loop is required.

The method consists of the following sequence of steps (Figure 4.7):



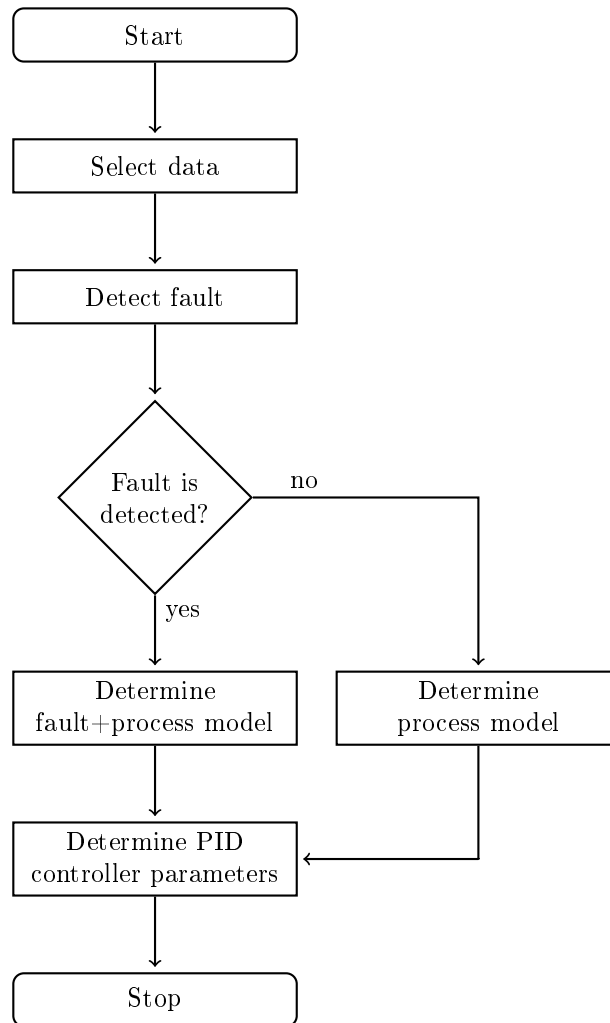


Figure 4.7: Proposed approach flowchart.

- i. Selection of a dataset.
- ii. Detect faults eventually present in control valve.
- iii. If no faults are detected, proceed with the mathematical model determination of the nominal process.
- iv. If faults are detected, proceed with the mathematical model determination of the process with the detected faults.
- v. Use the mathematical models determined in the previous step to find the new set of the controller parameters using numerical optimization techniques.

The method starts by selecting a set of operational closed-loop data that comprises the setpoint, manipulated, and controlled variables and that has sufficient dynamic information. This is followed by fault detection of the control valve using available detection methods. The proposed approach may be applied to several faults found in control valves. Stiction as the most common valve fault is detected in two distinctive steps. Firstly, an oscillation detection method is applied to data in order to detect the oscillatory behaviour, which is a typical indicator of the stiction phenomenon. If oscillatory disturbances are not detected, it is concluded that stiction is not present in the valve. However, if oscillations are detected, valve stiction detection is performed using methods based on signals shape, surrogate analysis, or system identification (see Section 3.2.4).

Then, the mathematical model of the process,  $f(\cdot)$  is determined using common modeling techniques, such as state-space models, transfer functions, and artificial neural networks. If faults in the control valve have been detected, the parameter estimation of the fault model,  $j(\cdot)$  is carried out. The parameter estimation of the models is based on the minimization of the weighted sum of the squared error between the controlled variable collected in loco  $\mathbf{y}_{\text{exp}}$ , and the corresponding mathematical model prediction  $\mathbf{y}$  as

$$\underset{\mathbf{p}_p}{\text{minimize}} \quad q_0 \sum_{i=1}^n (\mathbf{y}_{\text{exp}} - \mathbf{y})^2 \quad (4.33a)$$

subject to

$$\mathbf{x} = j(\mathbf{y}, \mathbf{u}_{\text{exp}}, \mathbf{p}_p) \quad (4.33b)$$

$$\dot{\mathbf{y}} = f(\mathbf{y}, \mathbf{x}, \mathbf{p}_p) \quad (4.33c)$$

where  $q_0$  is the weighting factor,  $\mathbf{u}_{\text{exp}}$  is the manipulated variable vector collected in loco,  $\dot{\mathbf{y}}$  is the time derivative of variable  $\mathbf{y}$ ,  $\mathbf{p}_p$  is the parameters vector of  $j(\cdot)$  and  $f(\cdot)$ , and  $n$  is the number of experimental points. This strategy is called direct closed-loop system identification (see Section 2.2.1) as it ignores the feedback loop and performs the estimation using only the manipulated and the controlled variables.

After the model has been identified, the tuning parameters of the PID controller are determined via the numerical minimization of the weighted sum of

the

- deviation of the controlled variable relatively to the setpoint variable,  $J_1$ , and
- control valve wear quantified by the movement degree of the moving parts of the valve,  $J_2$ ,

resulting in the objective function

$$J = q_1 J_1(\mathbf{y}_{sp}, \mathbf{y}) + q_2 J_2(\mathbf{u}), \quad (4.34)$$

with

$$J_1(\mathbf{y}_{sp}, \mathbf{y}) = (\mathbf{y}_{sp} - \mathbf{y})^\top (\mathbf{y}_{sp} - \mathbf{y}), \quad (4.35)$$

$$J_2(\mathbf{u}) = \sum_{i=1}^{m-1} (\mathbf{u}_{i+1} - \mathbf{u}_i)^2, \quad (4.36)$$

where  $q_1$  and  $q_2$  are weighting factors associated with  $J_1$  and  $J_2$ , respectively, and  $m$  is the number of simulation points. The PID controller parameters are determined by solving the mathematical problem

$$\underset{\mathbf{p}_c}{\text{minimize}} \quad J(\mathbf{y}_{sp}, \mathbf{p}_p, \mathbf{p}_c) \quad (4.37a)$$

subject to

$$\mathbf{x} = j(\mathbf{y}, \mathbf{u}, \mathbf{p}_p) \quad (4.37b)$$

$$\dot{\mathbf{y}} = f(\mathbf{y}, \mathbf{x}, \mathbf{p}_p) \quad (4.37c)$$

$$\mathbf{u} = h(\mathbf{y}, \mathbf{y}_{sp}, \mathbf{p}_c) \quad (4.37d)$$

$$\mathbf{y}_{LB} \leq \mathbf{y} \leq \mathbf{y}_{UB} \quad (4.37e)$$

$$\mathbf{u}_{LB} \leq \mathbf{u} \leq \mathbf{u}_{UB} \quad (4.37f)$$

$$\mathbf{p}_{c, LB} \leq \mathbf{p}_c \leq \mathbf{p}_{c, UB} \quad (4.37g)$$

$$g(\mathbf{p}_c) \leq 0, \quad (4.37h)$$

where  $\mathbf{p}_c$  is the parameters vector of the PID controller,  $h(\cdot)$  is the PID control law, and  $g(\cdot)$  is the functions set representing the conditions characteristic of the industrial process and enforcing additional criteria to the optimization problem

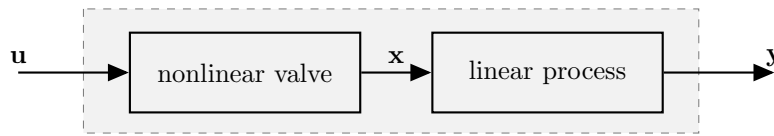


Figure 4.8: Hammerstein Model.

(for instance, limits in the overshoot or the decay ratio and limits associated to the process operation). Subscripts LB and UB refer to lower and upper bounds, respectively.

The approach described above was implemented in GNU Octave 3.8.1 using its general nonlinear minimization via `sqp()` successive quadratic programming solver. For problems (4.33) and (4.37), the solver stopping criterion was set to  $10^{-20}$  and  $10^{-5}$ , respectively, and the maximum number of iterations to  $10^6$ .

### 4.3.2 Discussion of results

In the following, the proposed approach is applied to a simulated process both in faulty and fault-free scenarios. Particularly, stiction will be considered as it is the most common valve fault found in the industry.

#### ■ Data selection

A plant simulator based on the Hammerstein Model frequently used to model processes with faulty valves (Figure 4.8) was used to generate two datasets: one for a healthy control loop and another with stiction in the control valve. A Hammerstein Model consists of a non-linear element in series with a linear dynamic part. In the present context, the non-linear element represents the faulty valve while the linear part models the process dynamics. In order to apply continuous optimization methods, a smoothed version of the Chen Model (Brásio et al., 2014) is used to model stiction ( $j(\cdot)$  in (4.33b)) and the state-space model

$$\dot{\mathbf{y}} = a(\mathbf{y} - \bar{\mathbf{y}}) + b(\mathbf{x} - \bar{\mathbf{x}}), \quad (4.38)$$

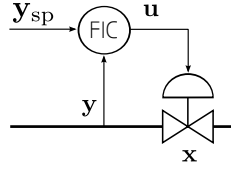


Figure 4.9: Flow control loop.

where  $a$ ,  $b$ ,  $\bar{y}$ , and  $\bar{x}$  are the state-space model constants to model the process dynamics ( $f(\cdot)$  in (4.33c)). The PID control law ( $h(\cdot)$  in (4.37d) is

$$\mathbf{u}_k = \mathbf{u}_{k-1} + k_C \left[ (\mathbf{e}_k - \mathbf{e}_{k-1}) + \frac{\Delta t}{\tau_I} \mathbf{e}_k + \frac{\tau_D}{\Delta t} (\mathbf{e}_k - 2\mathbf{e}_{k-1} + \mathbf{e}_{k-2}) \right], \quad (4.39)$$

where  $\mathbf{e}_k$  is the error signal,  $k_C$  is the proportional gain,  $\tau_I$  is the integral time (or reset time), and  $\tau_D$  is the derivative time. This algorithm is the digital parallel version of the PID controller using the velocity form (Seborg et al., 2010).

Closed-loop data from a flow control loop (Figure 4.9) was generated using the following parameters:  $a = -0.1 \text{ s}^{-1}$ ,  $b = 0.3 \text{ kg } \%^{-1} \text{ s}^{-2}$ ,  $\bar{y} = 0.0 \text{ kg s}^{-1}$ ,  $\bar{x} = 0.0 \%$ ,  $k_C = 1.6 \text{ \% s kg}^{-1}$ , and  $\tau_I = 10 \text{ s}$ . Following common industrial practice, no derivative action is used and, therefore,  $\tau_D = 0 \text{ s}$ . The Chen Model parameters ( $f_D, f_J$ ) were defined as  $(0, 0) \%$  for the healthy valve case and  $(5, 3) \%$  for the sticky valve case. The accuracy parameter of the smoothed version of the Chen Model was set to 5. Finally, a step excitation on the setpoint variable with amplitude of  $10 \text{ kg s}^{-1}$  is applied to the system to generate the response  $y$ . In the case of the healthy valve, a second excitation is performed with negative amplitude ( $-5 \text{ kg s}^{-1}$ ) in order to include sufficient dynamical information about the system. The obtained datasets contain  $n = 6001$  points covering an interval of 10 min with a sampling period of 100 ms. The experimental variables are pictured in the first column of Figures 4.10 and 4.11. Although variable  $\mathbf{x}_{\text{exp}}$  is also drawn, it is often not available in industrial environments.

### ■ Fault detection

The oscillation detection method described in Section 3.3 is applied to the data in order to detect oscillatory behaviour. Then, valve stiction detection is performed by the Yamashita's method (Yamashita, 2006a).

The results of the detection methods (Table 4.8) show that an oscillatory be-

Table 4.8: Fault detection results.

Dataset	Detection of				
	Oscillation		Stiction		
	Period	Evaluation	$\rho_1$	$\rho_3$	Evaluation
1	–	×	–	–	–
2	36.6 s	✓	0.969	0.811	✓

Data preprocessed by downsampling to 1 second.

×: not detected, ✓: detected.

Table 4.9: Process and fault+process modeling results.

	parameters $p_p$						quality	
	$a$ $s^{-1}$	$b$ $kg\ \%^{-1} s^{-2}$	$\bar{y}$ $kg\ s^{-1}$	$\bar{x}$ %	$f_D$ %	$f_S$ %	$J$ –	$R^2$ –
<i>Dataset 1</i>								
exp	-0.100	0.300	0.000	0.000	–	–	–	–
init	-1.000	1.000	0.001	0.001	–	–	$7.07 \times 10^6$	0.5463
fit	-0.099	0.297	0.001	0.005	–	–	$9.41 \times 10^2$	0.9995
<i>Dataset 2</i>								
exp	-0.100	0.300	0.000	0.000	7.000	2.000	–	–
init	-1.000	1.000	0.001	0.001	0.000	0.000	$4.51 \times 10^8$	0.3702
fit	-0.093	0.268	0.470	0.001	5.321	1.517	$1.01 \times 10^5$	0.9975
LB	-10.000	10.000	0.001	0.001	0.001	0.001	–	–
UB	10.000	10.000	10.000	10.000	10.000	10.000	–	–

Dataset 1:  $q_0 = 100\ s^2\ kg^{-2}$ . Dataset 2:  $q_0 = 1000\ s^2\ kg^{-2}$ .

haviour is only detected in the dataset 2 characterized by an oscillation period of 36.6 s. Consequently, only this set is analysed with the Yamashita's method that detects stiction since the values of  $\rho_1$  and  $\rho_3$  are above the limit of 0.25.

### ■ Process or fault+process modeling

The optimization problem described in (4.33) was used to the parameter estimation. Because stiction was not found in dataset 1, only the nominal process modeling will be performed in this dataset and, consequently,  $(f_D, f_J) = (0, 0)\%$  is considered in (4.33b). A fault+process model will be identified for dataset 2. The weighting factor  $q_0$  is set to  $100\ s^2\ kg^{-2}$  for dataset 1 and to  $1000\ s^2\ kg^{-2}$  for dataset 2.

The modeling results are presented in Table 4.9. It contains the values of pa-

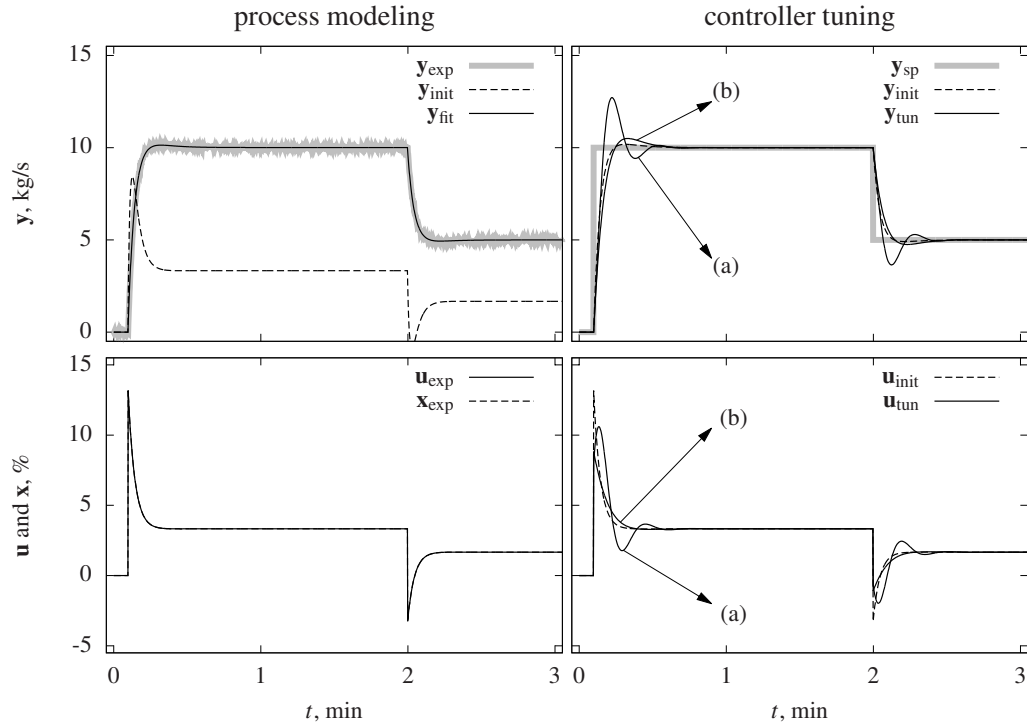


Figure 4.10: Dataset 1 results.

parameters  $p_p$  used for experimental data generation (exp), for optimization problem initialization (init), and those obtained from the curve fitting (fit). Also, the quality indicators, namely the objective function value,  $J$ , and the determination coefficient,  $R^2$ , are presented along with the lower and upper bounds.

The first column of Figures 4.10 and 4.11 illustrates the modeling results of Table 4.9. Both figures draw the experimental controlled variable in grey solid thick line. Even though the initial guesses are quite poor (black dashed line), the identification process is capable of capturing well the dynamics of the nominal process in dataset 1 and of the fault+process in dataset 2 (black solid line), reflected by the high values of  $R^2$  and the drastic objective function value reduction from the initial guesses to the final fits. The obtained parameter values are very similar to the ones used in the data generation.

### ■ PID controller tuning

The PID controller tuning is performed based on proportional and integral actions (configuration of the collected data) and, therefore, the decision variables

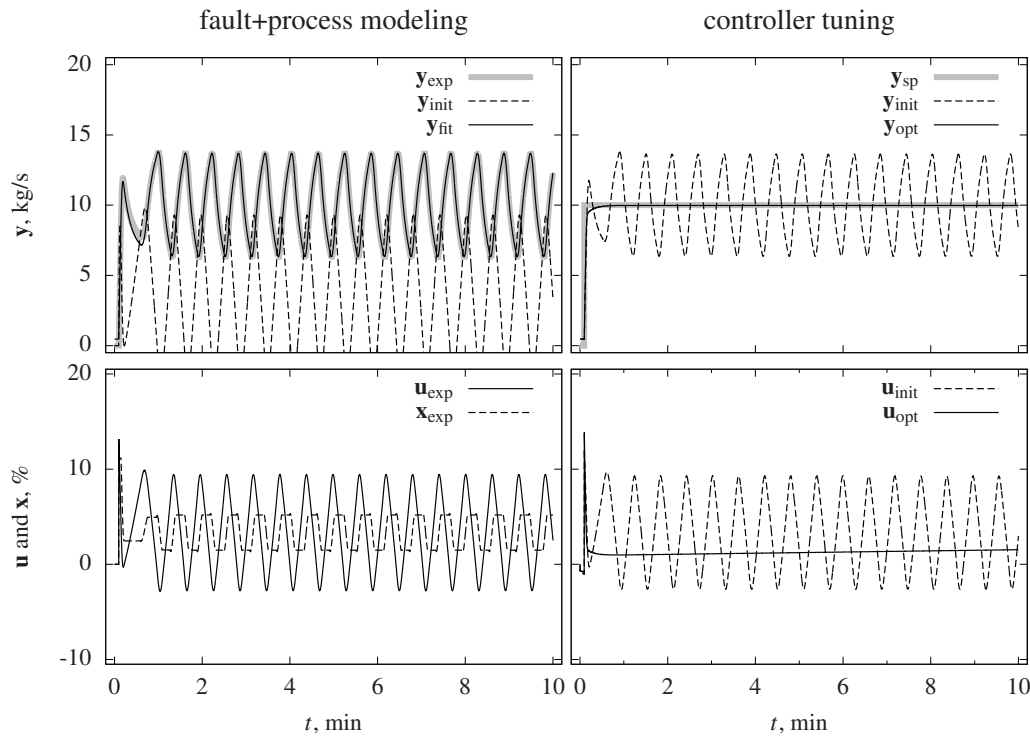


Figure 4.11: Dataset 2 results.

are the controller gain  $k_C$  and the integral time constant  $\tau_I$ . The tuning problem in (4.37) considers the models obtained in the previous subsection to define functions  $j(\cdot)$  and  $f(\cdot)$ , the setpoint variable excitation present in the experimental data to define  $y_{sp}$ , the weighting factors  $q_1 = 1 \text{ s}^2 \text{ kg}^{-2}$  and  $q_2 = 10 \text{ \%}^{-2}$ , and bounds in the parameters  $p_c$ .

The tuning results are presented in Table 4.10 where PID controller parameters  $p_c$  used to generate the experimental data (exp), to initialize the optimization (init), and the tuning results (tun) are indicated along with the process overshoot (OS), quality indicators and bounds. The second column of Figures 4.10 and 4.11 presents the tuning results. Here, the setpoint variable (in grey thick solid line), the initial response (in black dashed line) and the optimized response (in black solid line) are drawn for comparison. Also, initial and final manipulated variable responses are pictured in the bottom figure.

In case of dataset 1, two sets of parameters identified by (a) and (b) were determined. Set (a) was obtained without imposing any additional operational constraints. In comparison to the initial response, it reduced the aggressiveness of the



Table 4.10: PID controller tuning results.

	parameters $\mathbf{p}_c$			process charact.	quality	
	$k_C$ % s kg <sup>-1</sup>	$\tau_I$ s	$\tau_D$ s	OS %	$J$ –	$R^2$ –
<i>Dataset 1</i>						
init	1.300	8.000	0.000	1.83	$3.76 \times 10^3$	0.9382
tun (a)	0.744	1.651	0.000	27.15	$2.91 \times 10^3$	0.9207
tun (b)	0.868	6.165	0.000	5.00	$3.15 \times 10^3$	0.9154
<i>Dataset 2</i>						
init	1.300	8.000	0.000	19.33	$3.72 \times 10^4$	0.1310
tun	1.462	31.361	0.000	0.00	$3.23 \times 10^3$	0.8397
LB	0.100	0.001	0.000	0.00	–	–
UB	30.000	300.000	0.000	5.00	–	–

Datasets 1 and 2:  $q_1 = 1 \text{ s}^2 \text{ kg}^{-2}$  and  $q_2 = 10 \text{ \%}^{-2}$ .

control valve for a similar rise time and superior overshoot in the controlled variable. The superior overshoot (27.15 %) augments the deviation of the controlled variable from the setpoint which is evident in the reduction of  $R^2$  from 0.9382 to 0.9207. Nonetheless, the objective function  $J$  composed by two terms was clearly decreased from  $3.76 \times 10^3$  to  $2.91 \times 10^3$ . The first term  $J_1$  (closely related to the  $R^2$ ) became more prominent throughout the optimization process, justifying the reduction of  $R^2$ . However, the second term  $J_2$  related to the valve wear was drastically reduced, due to the less aggressive behaviour observed in the control valve. According to the selected weights, the control loop performance was greatly improved.

In some processes, an overshoot of this magnitude in the controlled variable influences considerably the final product quality. So, it is important to monitor and control the deviation of the controlled variable from the setpoint when setpoint changes occur. This criterion may be imposed in the proposed approach defining

$$g(\mathbf{p}_c) = \begin{bmatrix} \text{OS} & - & \text{OS}_{\max} \\ \text{OS}_{\min} & - & \text{OS} \end{bmatrix}, \quad (4.40)$$

in the condition (4.37h). Set (b) was determined considering this additional characteristic for  $\text{OS}_{\min} = 0 \text{ \%}$  and  $\text{OS}_{\max} = 5 \text{ \%}$ , which guarantees an acceptable range of the controlled variable overshoot. This new set of tuning parameters of the PID

controller generated a response with an overshoot of 5 % (the maximum value of the defined range), while the valve movement and the overall deviation from setpoint were penalized in the objective function. These parameters clearly improved both the pre-tuning loop behaviour and the loop response using parameters (a), because high overshoots and valve wear are avoided.

In the case of the dataset 2, the tuning did not consider the constraints on overshoot. Nevertheless, the pre-tuning behaviour of the control loop (see Table 4.10 and Figure 4.11) is clearly improved. The initially observed oscillations in the controlled and manipulated variables were totally removed, allowing the PID controller to obtain a clean response. This was achieved with an increase of the proportional gain and of the integral time.

# Chapter 5

## Soft sensing technology

This chapter concerns the issue of soft sensing technology as a way to generate new data that usually is not readily available from on-line instrumentation or laboratory measurements. The methodology, the data pre-processing, the techniques for soft sensing, and the concept drift detection and handling are deeply reviewed.

Because glycerine concentration process in an energy intensive process, the soft sensor technology has a high potential in the final glycerine quality prediction. In this chapter, the soft sensor technology is applied in the prediction of the glycerine quality in an industrial scenario for real-time monitoring and control purposes.

### 5.1 Importance and definition

In order to meet economic and environmental targets in the process industry, a continuous improvement of production processes has been pursued. This has been done in a very enriched data environment because, currently, industrial processes are highly instrumented with a large number of physical sensors providing a huge amount of data for process optimization. However, while data amount is increasing exponentially, the knowledge extraction through the understanding of that data remains a challenge.

Motivated by the advances of low-cost technology (such as computer hardware, graphical user interfaces and high-level software packages), the stored data

has been used increasingly to obtain new knowledge by building virtual instrumentation, called soft sensors. Soft sensors present an attractive feature in the industrial environment because they are low cost alternatives compared with the expensive hardware devices. Furthermore, they may operate in parallel with the existing instrumentation giving useful information for the detection of faults in the process and in the instrumentation. The soft sensors may be easily developed and implemented on the existing hardware with little effort and high transparency, and may be retuned when conditions and, consequently, parameters change. In addition, the estimation of product quality variables, one of common soft sensor applications, in real time reduces the long time delays typical of some physical sensors, such as the chromatographs. Soft sensors are a valuable tool in many application fields. Some industrial applications are found in refineries, chemical plant, cement kilns, power plants, pulp and paper industry, food processing, nuclear power plants, and urban and industrial waste processing plants.

Their functions include measuring system backup, what-if analysis, prediction in real time for process control, sensor validation and fault diagnosis, as Fortuna et al. (2007b) explains.

Due to their practical utility and a priori knowledge absence, soft sensors have enjoyed an increasing popularity (Shang et al., 2014). They are based on a wide variety of methods employed in the development of soft sensors, such as the commonly used system identification methods, machine learning methods and data mining methods. Since system identification methods were already addressed in Chapter 2, some concepts related to machine learning and data mining fields will be detailed in the present chapter.

The term soft sensor emerged from the combination of the words *software* and *sensors* and refers to the estimation in a computer program of any process variable (usually product quality related variables) by using mathematical models and data acquired from the physical sensors. Others definitions are found in literature that also translate this concept as inferential sensors (Khatibisepehr et al., 2013), virtual on-line analysers (Komulainen et al., 2004), software sensors (Soons et al., 2008), and observer-based sensors (Pierri et al., 2008).

Soft sensors may be classified into two types: the model- and the data-driven soft sensors. The model-driven soft sensors are based on first principles models

translating the physical and chemical background of the process. Their main drawback is the fact that the process expert knowledge is required in order to build the model. Besides, oftentimes the theoretical background is not sufficient because the real process is significantly influenced by others factors not accounted in the model. Nonetheless, model-driven soft sensors are the most used type in inferential control. Some examples of these sensors may be found in Welch and Bichop (2001), Prasad et al. (2002), Frau et al. (2009), Frau et al. (2010), and Boizot et al. (2010).

Due to the fact that the data-driven soft sensors do not have the complexity problems inherent to the model-driven approach, the former have gained popularity because they use directly plant data. The most used data-driven modelling techniques are the principal component analysis, partial least squares, support vector machines, artificial neural networks, fuzzy systems, time-series models, and hybrid models. The following sections will mainly focus on data-driven methods.

## **5.2 State-of-the-art**

This section provides an overview of the techniques used for soft sensor development and use, with a particular focus on the challenges and solutions encountered in the process industries. Other relevant reviews about the topic may be found in the literature (Mansano et al., 2014; Saptoro, 2014; Haimi et al., 2013; Kano and Fujiwara, 2013; Luttmann et al., 2012; Escobar, 2012; Kadlec et al., 2011; Slišković et al., 2011; Li et al., 2011; Kadlec et al., 2009; Fortuna et al., 2007b, 2005b; Gonzalez, 1999).

### **5.2.1 Development methodology**

The availability of industrial data is a factor that motivates wider interest in soft sensor development and use. However, special care should be taken because the application of standard data-driven modelling methodologies for soft sensors design may lead to model degradation due to several aspects, such as the contamination of data by outliers. Therefore, a systematic procedure for the soft sensors development should be followed (Lin et al., 2007).

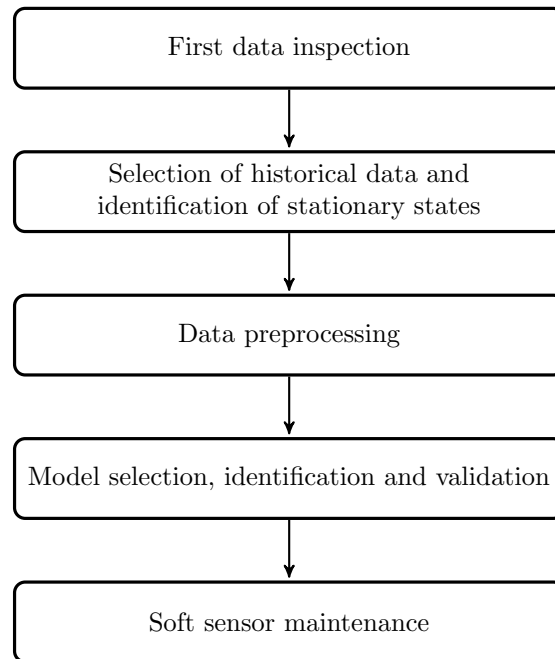


Figure 5.1: Methodology for the soft sensors development.

A rather general methodology, applicable to batch and continuous processes (Figure 5.1) comprises the following steps:

**Data inspection:** The initial data inspection is performed with the aim of obtaining their structure and identification of eventual problems in the data, namely collinearity, outliers, missing values, sampling frequencies, measurements delays, and noise. These issues are reviewed in Fortuna et al. (2007b) and Kadlec et al. (2009). In this stage, it is possible to evaluate if the requirements of the system identification are accomplished and to do a reasonable decision of the method that will be used.

**Historical data selection:** The selection of the data for the identification and evaluation of the model is an important task. The stationary parts must be identified and selected. This task is usually performed by manual annotation of the data.

**Data pre-processing:** The role of the data pre-processing is to transform the data to be more efficiently processed in the system identification step. The problems detected in data at Data Inspection stage are handled with the

appropriated tools. Fortuna et al. (2007b) and Kadlec et al. (2009) discuss some methods that may solve those problems. Other techniques to deal with relevant topic regarding collected data quality and its pre-processing are summarized in Section 5.2.2.

Due to the characteristics of the industrial data, the data pre-processing is a critical stage because, at the moment, it requires a large amount of manual work and expert knowledge about the underlying process.

**System identification:** In soft sensing, system identification comprises model selection, identification, and validation (see Chapter 2). The selection of the model is crucial for the performance of the soft sensor because the model is its engine. Currently, there is no unified theoretical approach for the model selection and, consequently, it has been done based on the experience and preference of the developer. The most common approach for the model selection is to start with a simplified model structure and to assess its fitting to the data. Then, the complexity is increased as long as there is a significant improvement of the performance. After the model selection and identification, the obtained soft sensor is validated with independent data. The mean squared error is the most popular method to assess the fitting quality. Often, visual tools, such as the four-plot analysis (NIST and SEMATECH, 2012; Fortuna et al., 2007b), are also used to compare the predictions with the real data. Nevertheless, the resulting performance is dependent on the subjective judgement of the model developer (Fortuna et al., 2007b).

**Maintenance:** The final step of the methodology is the maintenance of the soft sensor. The existence of changes of the process or of external process conditions may affect the process state, as well as the data in terms of variance and mean. Some examples of such changes are varying environmental conditions, purity of the input materials and catalyst deactivation. Therefore, in these cases, the recalibration of the measurement devices or the adaptation of the soft sensor must be carried out.

Currently, most of the soft sensors do not include any automated maintenance mechanism. Furthermore, there is often no objective measure for assessing the quality level of the soft sensor. Instead, it is dependent on the

model operator subjective perception based on visual interpretation of the deviation between the correct target value and its prediction. This motivated the use of adaptive versions of the methods used in the development of soft sensors, such as moving window PCA (Wang et al., 2005), moving window PLS (Ni et al., 2014), recursive PCA (Li et al., 2000) among others (see Kadlec et al. (2009)).

### 5.2.2 Data pre-processing

A careful analysis of the available laboratory and operational data enables to select relevant variables and to assess data quality in order to extract important information about the process. The experience and expertise of those involved in the daily operation provides a valuable knowledge about the plant and may help in the determination of the most relevant process variables and of the hardware sensors performance (Khatibisepehr, 2013).

Kadlec (2009) classifies the data into two main classes: the historical and the real-time data. Historical data describe the process behaviour in the past. Usually, modeling tasks (model selection, training and validation, and parameter optimization) use batches of historical data that allow to take into account the delays between the input and the output variables. Indeed, using historical data, it is possible to compensate the delay between measurements by entering the values of the output variables at the time of taking the sample.

In contrast, in real-time data the delay between measurements may not be compensated because the data is arriving in an incremental way. This restricts their application to real-time simulation, adaptivity, and control. Input variables are used to simulate the process and, when the output variables are available, they are used to evaluate the model performance during the on-line prediction phase. If a performance deterioration is detected, an adaptation mechanism is activated using historical data (Kadlec, 2009).

Industrial plants are heavily instrumented for process control purposes and, consequently, the recorded data is composed by a large number of variables. In such scenario, it is likely that some problems arise in the data. Several works have analysed industrial data problems and developed approaches for handling irregular datasets (Khatibisepehr and Huang, 2008). Figure 5.2 shows some issues



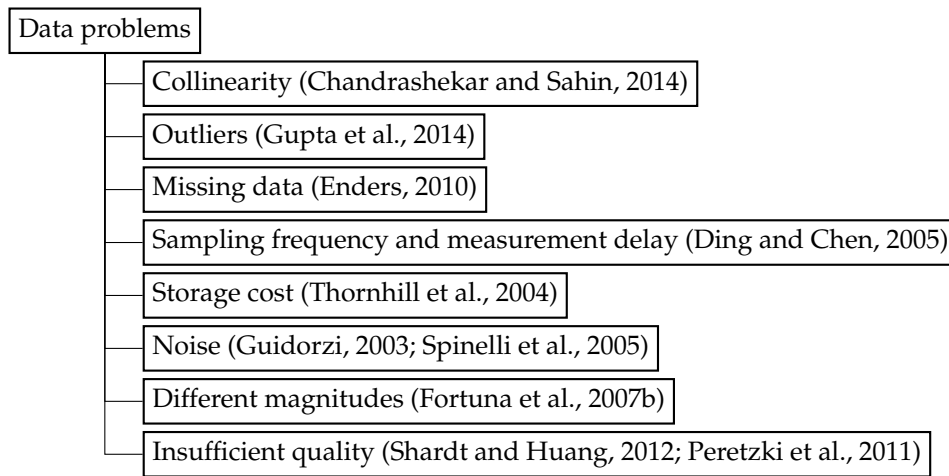


Figure 5.2: Problems present in industrial data.

that usually occur in data and enumerates some works that describe ways to handle them. The next sections will focus on these problems of the data.

### ■ Collinearity

For safety and process control purposes, it is common to have redundant or closely related measurement in process plants. However, this may result in the presence of collinearity in the measured data (for instance, two neighbour temperature sensors in a distillation column will collect data that is strongly correlated). Based on the classification of Chandrashekar and Sahin (2014), the data collinearity may be handled by using the dimension extraction and the features selection methods (Figure 5.3).

Dimension extraction methods describe a large set of data accurately generating a reduced number of variables. The most famous methods are Principal Component Analysis (Zamprogna et al., 2005) and Partial Least Squares.

Usually, a subset of variables may efficiently describe the input data. Usually, the choice of important variables is done via manual selection by system experts. However, this task may not be feasible for large and highly integrated processes (Warne et al., 2004). Several techniques were developed to reduce irrelevant and redundant variables helping to understand data, to reduce computation requirements, and to improve the soft sensor performance. These techniques are usually called feature selection methods (Chandrashekar and Sahin,

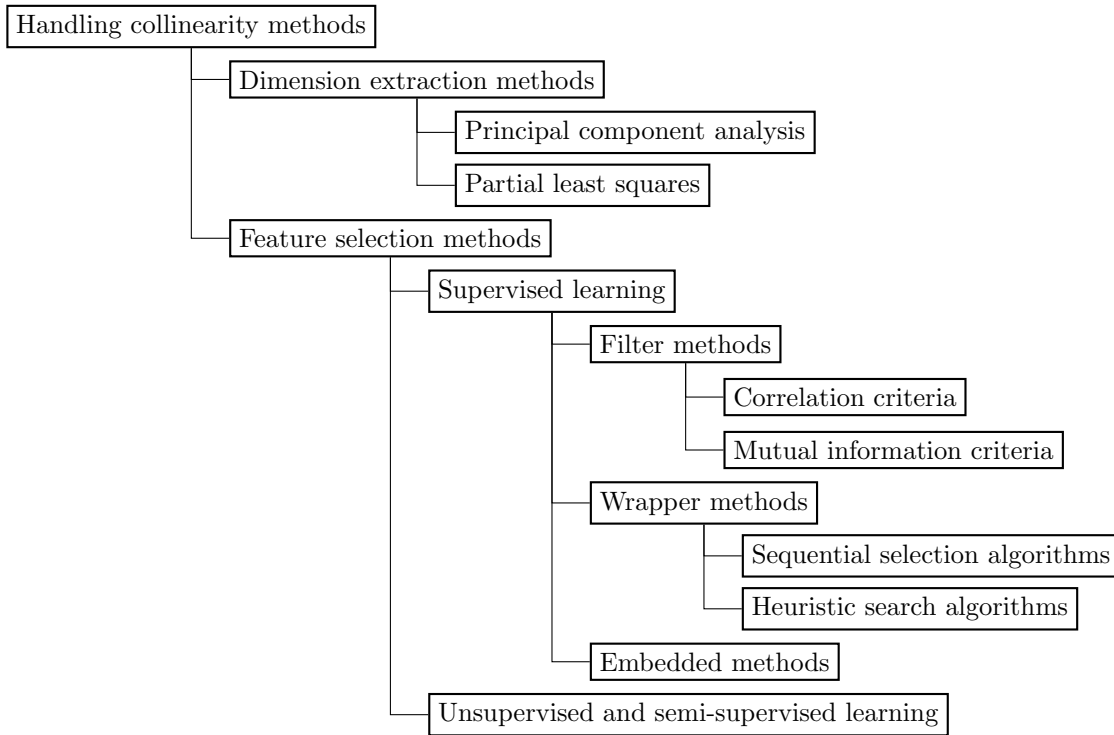


Figure 5.3: Methods for handling collinearity in data.

2014; Guyon and Elisseeff, 2003).

In the context of soft sensing, the most relevant feature selection methods are the supervised learning algorithms which include the filter, the wrapper, and the embedded methods that are used to find the subset of useful variables based on a pre-selected criterion.

Filter methods use the selection criterion to rank the variables and select them based on a predefined threshold value. The most used criteria are the Pearson correlation and the mutual information that measure the dependency between two variables. Considering a one-dimensional space, the former is defined as (Chandrashekar and Sahin, 2014)

$$R(i) = \frac{\text{cov}(X, Y)}{\sqrt{\text{var}(X) \cdot \text{var}(Y)}}, \quad (5.1)$$

where  $\text{cov}(\cdot, \cdot)$  is the covariance and  $\text{var}(\cdot)$  the variance of the specified variables. Varying between -1 and 1, the Pearson correlation measures linear associations between input variables  $X$  and output variables  $Y$ . A value of zero means that

the two variables are independent. Contrarily, the later calculates a non-linear measure of the dependency between  $Y$  and  $X$  by (Chandrashekar and Sahin, 2014)

$$MI(Y, X) = H(Y) - H(Y|X) \quad (5.2)$$

where  $H(Y)$  is the entropy of  $Y$  and  $H(Y|X)$  is the conditional entropy of  $Y$  observing  $X$ . In addition, the mutual information may be expanded for multidimensional spaces (Souza and Araujo, 2011). On one hand, these methods are advantageous because they are simple, computationally light, avoid over-fitting, and work well for certain datasets. On the other hand, the obtained subset may not be optimal because, although the selected variables are uncorrelated with the output variable, they may be also correlated to other input variables. Compared to the Pearson correlation, mutual information is advantageous because it may deal with non-linearities and with a large number of variables (Meyer, 2008).

Wrapper methods use the predictor performance as a selection criterion, wrapping the predictor on a search algorithm which will find the subset with the higher performance. The main disadvantage is the necessity of creating a new model (soft sensor) for each subset evaluation. This greatly increases the number of computations.

Embedded methods aim to reduce the computation time required by the wrapper methods incorporating the feature selection method as part of the training process. Basically, one of the selection criteria used by filter methods is incorporated into the objective function to simultaneously maximize the correlation between the selected inputs and the output, and to minimize the correlation between the selected inputs.

## ■ Outliers

An outlier is an observation that deviates markedly from other observations of the time series. The identification of outliers is a critical part of the data pre-processing because outliers have a negative effect on the model performance (Di-Bella et al., 2007). Outliers may arise from hardware failures, incorrect readings from instrumentation, transmission problems, and *strange* process working conditions (Kadlec et al., 2009).

Outlier detection may be performed either by formal tests (also known as tests

of discordance) or by informal tests (also called labelling methods) (Seo, 2006). Although formal tests usually require test statistics based on the distribution assumptions and a hypothesis to determine if the extreme value is an outlier of the distribution, they are quite powerful under well-behaving statistical assumptions. Contrarily, the informal tests are simpler to apply. Some examples are the standard deviation method, the  $z$ -score method, the modified  $z$ -score method, the Tukey method (boxplot), the adjusted boxplot, the  $MAD_E$  method, and the median rule. When it is difficult to identify the data distribution or transform it into a proper distribution, these methods may falsely identify outliers. However, they may be used for a first detection of outlier points (Seo, 2006). Comparisons of the different approaches are provided in the literature (Penny and Jolliffe, 2001; Matsumoto et al., 2007; Seo, 2006; Manoj and Senthamarai, 2013).

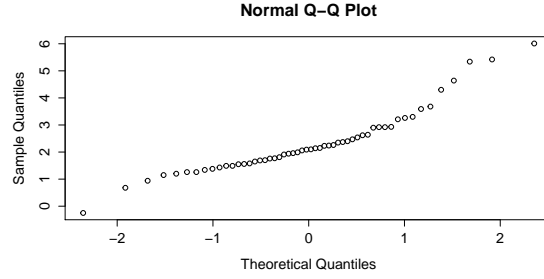
Some formal methods to handle outliers focus on univariate time series that follows an approximately normal distribution. As stated by NIST and SEMATECH (2012), Grubbs, Tietjen-Moore and generalized Extreme Studentized Deviate (ESD) tests are the three most commonly used outlier detection tests and are based on the criterion of distance from the mean. The test developed by Grubbs (1969) and Stefansky (1972) is used to detect single outliers in a univariate time series. Later, it was extended by Tietjen and Moore (1972). The main limitation of these two tests is that they must specify exactly the number of outliers. Contrarily, the generalized ESD test developed by Rosner (1983) only requires that an upper bound for the suspected number of outliers is specified. Given the upper bound  $r$ , the generalized ESD test essentially performs  $r$  individual tests considering the two hypothesis:

$H_0$ : There are no outliers.

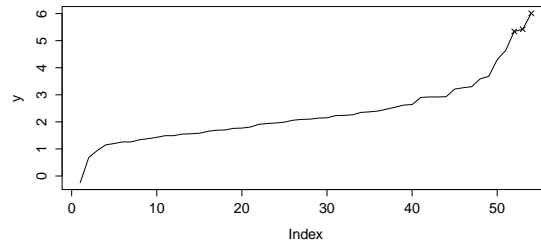
$H_1$ : There are up to  $r$  outliers.

For a given test  $i$ , the statistic  $R_i$  and the respective critical value  $\lambda_i$  ( $i = 1, \dots, r$ ) are computed and compared, considering the following definitions:

$$R_i = \frac{\max_i |y_i - \bar{y}|}{s}, \quad (5.3)$$



(a) Data distribution.



(b) Outliers identification.

Figure 5.4: Outliers detection of the Rosner (1983) dataset.

and

$$\lambda_i = \frac{(n-i)t_{p,n-i-1}}{\sqrt{(n-i-1+t_{p,n-i-1}^2)}}(n-i+1), \quad \text{with } p = 1 - \frac{\alpha}{2(n-i+1)}, \quad (5.4)$$

where  $\bar{y}$  and  $s$  denote the sample mean and the sample standard deviation, respectively,  $t_{p,\nu}$  is the  $100p$  percentage point from the  $t$ -distribution with  $\nu$  degrees of freedom,  $\alpha$  is the significance level, and  $n$  is the number of observations. The exact number of outliers is determined by finding the largest  $i$  such that  $R_i > \lambda_i$  for a  $\alpha$  level of confidence.

Other outlier detection procedures found in literature are the  $3\sigma$  edit rule and the Hampel identifier. The  $3\sigma$  edit rule (Ratcliff, 1993) is a standard deviation method based on  $\sigma = 3$  and consists on

$$|x(i) - \bar{x}| > t \sigma, \quad (5.5)$$

where  $\bar{x}$  is the mean of the data sequence and  $t = 3$  is the threshold. According to Pearson (2005), it is one of the most popular approaches to outlier detection

and is based on the statistical parameters mean and standard deviation. The basic difficulty of the method is the sensitivity of the statistical parameters to outliers presence which tends to mask the outliers.

An also very cited method is the Hampel identifier (Davies and Gather, 1993) that substitutes the mean and the standard deviation of the  $3\sigma$  edit rule by less sensitive parameters. The obvious alternatives are the median to replace the mean and the median absolute deviation from the median (MAD) to replace the standard deviation. The MAD scale estimative is defined as

$$\text{MAD} = 1.4826 \text{ median}(|x(i) - x^*|), \quad (5.6)$$

where  $x^*$  is the median of the data and the factor 1.4826 is the MAD corresponding to the standard deviation for normally distributed data. The method is more effective than the  $3\sigma$  edit rule but the MAD is identically zero if more than 50% of the observation have the same value. This behaviour causes a bad performance of the Hampel identifier. The combination of a moving window filter with the Hampel identifier is able to overcome this problem.

A recent review of Gupta et al. (2014) enumerates and classifies several more advanced approaches to handle outlier points.

### ■ Missing data

Most of the techniques for soft sensor development are not able to deal with missing data, that is, the data points that do not represent correctly the real process state and usually assume values like  $\pm\infty$  or 0. The most common causes of missing data are the hardware sensor failure, maintenance or removal. Other causes are related to the data transmission between sensors and databases, databases access, and other database errors (Kadlec et al., 2009).

Rubin (1976) introduced a widely used classification of the missing data handling methods. Let  $X = \{X_{\text{obs}}, X_{\text{miss}}\}$  denote the data matrix including both observed,  $X_{\text{obs}}$ , and unobserved/missing variables,  $X_{\text{miss}}$ , and let  $M$  denote the missingness indicator matrix expressing whether a particular variable is observed ( $M_{i,j} = 1$ ) or unobserved/missing ( $M_{i,j} = 0$ ). The classification considers three incompleteness mechanisms and describes how the probability of missing values

relates to data,  $p(M|X)$ :

- Missing completely at random (MCAR) data. The probability that an element is missing is independent of both the observed and the missing data and the missingness indicator is purely haphazard. The conditional distribution of  $M$  given  $X$  is given by

$$p(M|X) = p(M). \quad (5.7)$$

- Missing at random (MAR) data. The probability that an element is missing depends on the observed data only and

$$p(M|X) = p(M|X_{\text{obs}}). \quad (5.8)$$

In the present context, MAR data usually occurs due to planned missingness, i.e., some measurements may be more costly than others and are therefore obtained only for selected samples.

- Missing not at random data (MNAR) data or non-ignorable. The probability that an element is missing depends on the observed and unobserved data and (Enders, 2010)

$$p(M|X) = p(M|X_{\text{obs}}, X_{\text{miss}}). \quad (5.9)$$

In the present context, MNAR data usually occurs when some values are below the detection or quantification limit.

Figure 5.5 shows the influence of the different missingness mechanisms in the probability distribution of a variable  $A$ . The distribution of the variable is drawn using a solid line and circles. The three missingness mechanisms were applied to the variable and the resulting probability distributions are shown using dotted, pointed and solid lines. Subjected to missing data, the new distributions clearly differ from the original. This is specially evident for the MNAR mechanism whose distribution presents a bigger mean deviation.

In order to verify which of these mechanisms is present in data, it is necessary to perform an evaluation. Enders (2010) describes two tests for assessing

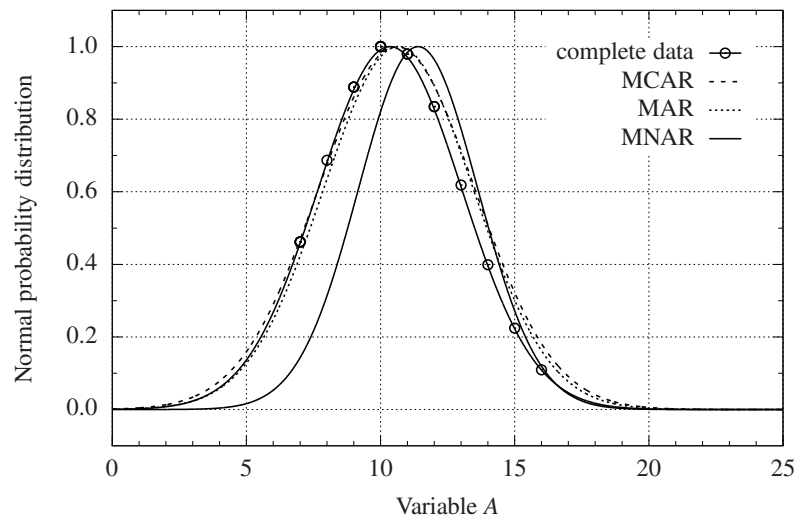


Figure 5.5: Influence of the missing mechanisms on the probability distribution.

MCAR data mechanism introduced by Dixon et al. (1988) and Little (1988). The univariate  $t$ -test based approach developed by Dixon et al. (1988) is a simple test for assessing MCAR mechanism using a  $t$ -test to compare data subgroups. Little (1988) proposed a multivariate extension of the univariate  $t$ -test based approach (see the algorithm in Appendix C.1) to evaluate simultaneously mean differences on every variable in the dataset, the multivariate Little MCAR test. Contrarily to MCAR, there is no way to confirm that the probability of missing data is solely a function of the observed variables (MAR mechanism). This represents a practical problem because two of the most recommended techniques to handle missing data (maximum likelihood method and multiple imputation) assume a MAR mechanism. There is also no way to verify the MNAR mechanism without knowing the values of the missing values.

Several methodologies to address the missing data problem were proposed (see Figure 5.6). These methods deal with missing data by removing the cases with incomplete scores or by filling in the missing values.

Case-wise deletion methods are the most common missing data handling approaches. List-wise deletion method (also known as complete-case analysis) discards the data for any case that has one or more missing data. The pair-wise deletion method (also known as available-case analysis) allows to use more of the data. It uses the available scores of the case with missing data to apply the



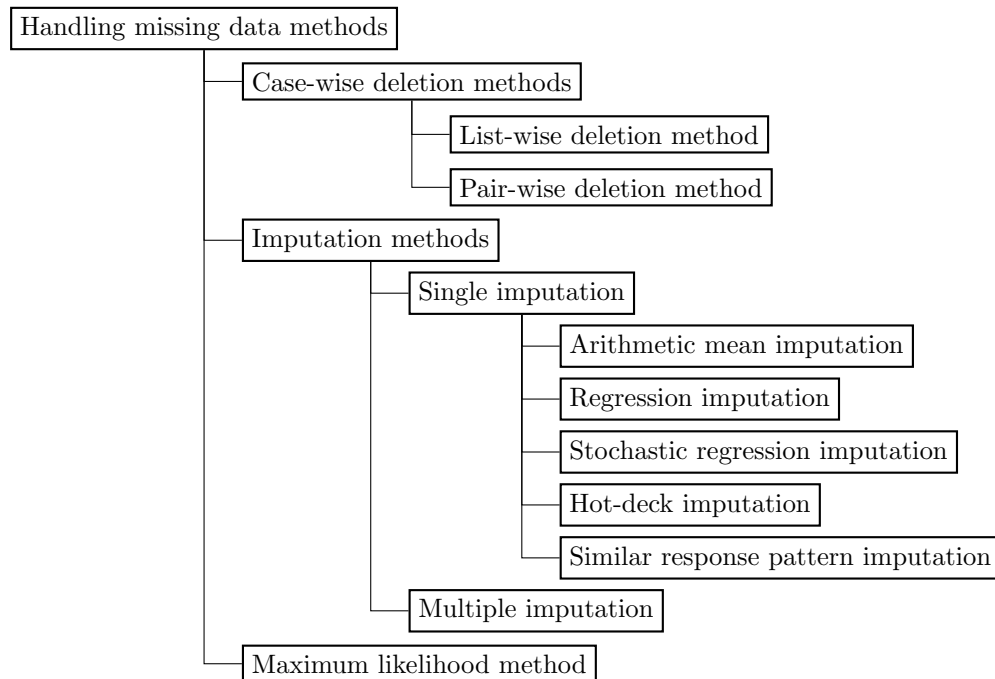


Figure 5.6: Methods for handling missing data.

techniques only to those scores. These methods are of easy implementation and are usually standard methods in statistical software packages. However, the removal of incomplete samples is not beneficial because it may lead to a considerable loss of information which can negatively impact the performance of the soft sensor. Essentially, these approaches assume the MCAR missing data mechanism producing distorted estimations when this assumption does not hold. Even, when the mechanism is valid, the deletion may reduce the data power. Also, the replacement of the missing values with a single statistical measure distorts the statistical distribution of the data.

Single and multiple imputation methods are attractive because they yield complete datasets. While the single imputation methods generate a single replacement value for each missing score, the multiple imputation methods create a set of points to be replaced in the missing values. But the imputation methods may produce biased parameter estimates and increase standard errors.

The single imputation methods category includes the following methods. Arithmetic mean imputation consists in filling the missing scores with the arithmetic mean of the available cases. Regression imputation replaces the missing scores

with values predicted from a regression equation. Stochastic regression imputation, which is a method similar to the previous one, adds a normally distributed residual term to the regression equation values. The hot-deck imputation replaces each missing score by the respective variable from a random observed case. Finally, the similar response pattern imputation replaces the missing value with the score from another case which has a similar response on a set of variables.

The aforementioned methods are traditional missing data handling methods usually encountered in literature reviews and statistical software packages. Maximum likelihood and multiple imputation methods are reported to be more efficient (Walczak and Massart, 2001; Khatibisepehr, 2013). They are essentially based on statistical principle methods which include explicit assumptions about the incompleteness of the data. Their algorithms are summarized in Appendix C.

### ■ Sampling frequency and measurement delay

The availability of several sources of information brings the possibility of having data with different sampling frequencies (rate). For instance, a simple dual-rate system is found in a discrete-time case in which the control updating period  $\Delta t_1$  is not equal to the output sampling period  $\Delta t_2$  ( $\Delta t_1 \neq \Delta t_2$ ). Multiple sampling frequencies are abundant in industrial processes mostly due to sensor and actuator speed constraints. In the cases of composition, density or molecular mass distribution control, these quality measurements are typically obtained after several minutes of analysis, while the manipulated variables could be adjusted at a relatively fast rate. Usually, sampling frequency is handled by synchronising the data (Kadlec and Gabrys, 2007) that consists of recording new samples only if one of the observed variables changes more than a pre-defined threshold value. Ding and Chen (2005) suggest to solve the multiple sampling frequencies problem by mapping the relationships between the available multi-frequency input and output data using the polynomial transformation technique and the lifting technique and by estimating the inter-sample (missing) output samples with the obtained model.

Closely related to data sampling frequency are the measurement delays of the process. Specially associated with laboratory measurements, the delays may reach up to several hours. Often, dealing with measurement time delays includes

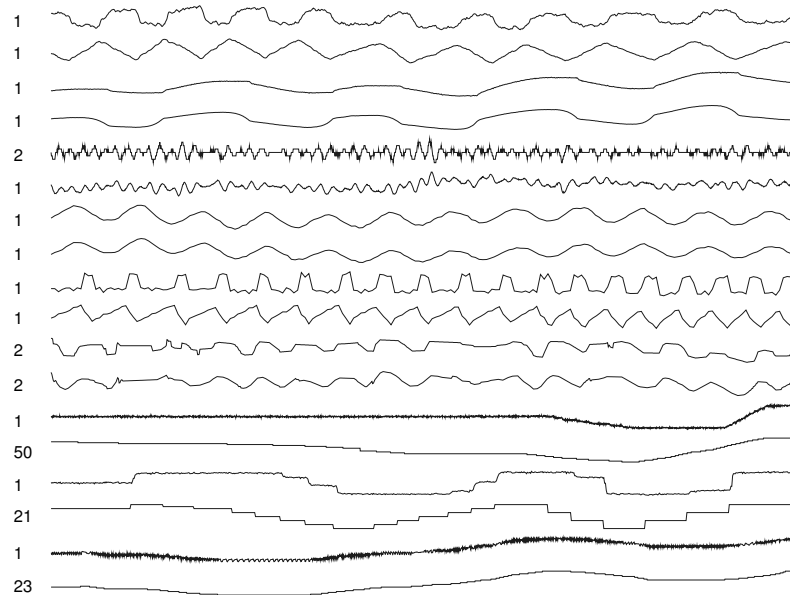


Figure 5.7: Industrial datasets of Jelali and Huang (2013) with different degrees of compression identified at the right  $y$ -axis.

a delayed version of the input variables into the feature selection (Kadlec and Gabrys, 2007; Souza et al., 2010).

### ■ Storage cost

Storage cost reduction is an often adopted policy in industrial plants. Usually, this policy has a negatively high impact because the stored data are a resource for valuable data-driven methods such as perform statistical monitoring, process control, fault detection and soft sensors development (Fortuna et al., 2007b; Thornhill et al., 2004). Commonly, storage cost reduction is achieved by the compression of data before the plant historian archives them (Figure 5.7). Compression techniques are mainly divided into piecewise linear and transform compression approaches. Mah et al. (1995) and Watson et al. (1998) compared the various compression techniques stating that the most effective way to compress large sets of industrial data is by transformation (Laplace, Fourier or wavelet) and by thresholding the insignificant transform coefficients. The piecewise linear compression technique showed to be less effective. Despite the results shown

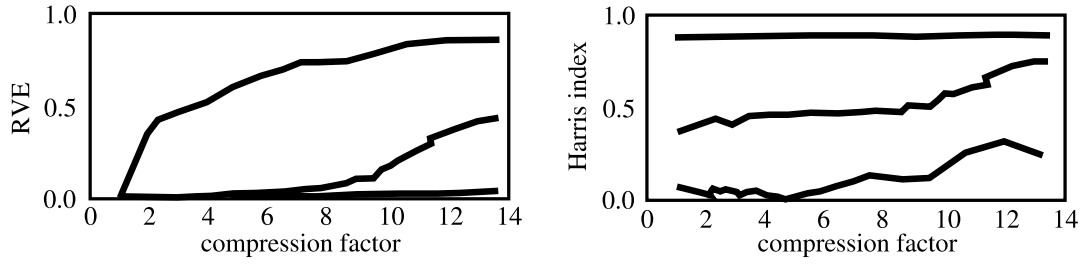


Figure 5.8: Effect of compression on data-driven analysis for different datasets (adapted from Thornhill et al. (2004)).

in these studies, plant historians usually use piecewise linear compression techniques (also known as direct techniques) because they may be applied in real-time environments to local data (Thornhill et al., 2004).

Thornhill et al. (2004) proved the negative impact of data compression in the data-driven methods results (statistical measures and control loop performance assessment). Figure 5.8 shows that the ratio between the variance of the original data  $y$  and the variance of the reconstructed data  $\hat{y}$  ( $RVE = \sigma_y^2 / \sigma_{\hat{y}}^2$ ) and the Harris index severely diverge when the compression factor is augmented. After the reconstruction of a compressed dataset, the data is composed by linear portions and their second derivative is zero everywhere apart from the places where linear segments join.

Therefore, the compression detection is an important procedure to perform during the pre-processing stage of a data-driven analysis. Thornhill et al. (2004) suggested the calculation of the compression factor defined by

$$\text{compression factor} = \frac{N}{m}, \quad (5.10)$$

where  $N$  and  $m$  are the number of samples of the original data and of the compressed data. Counting zero-valued second derivatives

$$\Delta(\Delta\hat{y})_i = \frac{\hat{y}_{i+1} - 2\hat{y}_i + \hat{y}_{i-1}}{h^2}, \quad (5.11)$$

where  $\hat{y}$  is the reconstructed signal and  $h$  is the sampling interval, gives a lower bound for the compression factor. Besides, the authors highlight the importance of the reconstruction of the data at the original sampling interval for an accurate assessment of the compression factor.

## ■ Noise

Industrial data is susceptible to short variations due to noise. Noise associated with measurements may arise from the sensors, the electrical equipment, or the process itself. The noise induced by the process may have origin in variations resulting from incomplete mixing, turbulence, and non-uniform multiphase flows (Seborg et al., 2010; Verhaegen and Verdult, 2012).

Filtering a signal consists in the suppression of some unwanted components from the signal. There are analog and digital filters (Seborg et al., 2010). The first ones play an important role in the removal of electrical noise (analog signal), specially in the fields of telecommunications. In order to damp out the electric noise, it is often used a low-pass filter described by the first-order differential equation

$$\tau_F \dot{y}_F(t) + y_F(t) = y_m(t), \quad (5.12)$$

Digital filters are applied to digital signals, usually found from process sensors. The digital version of the low-pass filter is given by

$$y_F(k) = \alpha y_m(k) + (1 - \alpha) y_F(k - 1), \quad (5.13)$$

where  $\alpha$  refers to the filter time constant (when  $\alpha = 1$ , no filtering is applied, and when  $\alpha \rightarrow 0$ , the measurement is ignored). Applying this filter may also be called as single exponential smoothing or exponentially weighted moving average (EWMA) filtering.

The double exponential filter (also known as second-order filter) is also very useful and is specially advantageous in dealing with signal drifts and in filtering high-frequency noise. Equivalent to two-series low-pass filter, the second-order filter is given by

$$y_F(k) = \gamma \alpha y_m(k) + (2 - \gamma - \alpha) y_F(k - 1) - (1 - \alpha)(1 - \gamma) y_F(k - 2), \quad (5.14)$$

where the second filter time constant  $\gamma$  may be assumed as  $\gamma = \alpha$  resulting

$$\bar{y}_F(k) = \gamma^2 y_m(k) + 2(1 - \alpha) \bar{y}_F(k - 1) - (1 - \alpha)^2 \bar{y}_F(k - 2). \quad (5.15)$$

**■ Different magnitudes**

Industrial data have different magnitudes depending on the variable units and on the process nature. This characteristic may hinder system identification because variables with larger magnitudes may be dominant over variables with smaller ones. In order to deal with, data scaling may be applied. The most common scaling methods are the min-max and the  $z$ -score normalizations defined by (Fortuna et al., 2007b)

$$\text{Min-max normalization: } x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}}(x'_{\max} - x'_{\min}) + x'_{\min}, \quad (5.16)$$

$$\text{z-score normalization: } x' = \frac{x - \bar{x}}{\sigma_x}, \quad (5.17)$$

where  $x$  and  $x'$  are the unscaled and scaled variables, the min and max subscripts refer to the minimum and maximum of the variable, and  $\bar{x}$  and  $\sigma_x$  are the mean and the standard deviation of the variable. When outlier presence is likely, the more robust  $z$ -score normalization approach is commonly preferred.

**■ Insufficient quality**

The main purpose of data quality assessment techniques is to determine whether the data contains sufficient excitation or information to be used in the system identification step given the model structure (Shardt and Huang, 2012).

Shardt and Huang (2012) present a framework for assessing the quality of routine operating data for system identification divided into two steps: the model segmentation via signal entropy and the data quality assessment via the Fisher information matrix.

Peretzki et al. (2011) develop an algorithm that searches and marks intervals suitable for process identification. It is a simple and efficient recursive algorithm that requires a minimum of process knowledge. Essentially, the steps are the search for excitation of the input and output, followed by the estimation of a Laguerre model combined with a chi-square test to check if at least one estimated parameter is statistically significant. The use of Laguerre models is crucial to handle processes with dead-time without explicit delay estimation. The method was tested with a three year dataset from more than 200 control loops. It was able

to find all intervals in which known identification experiments were performed additional intervals with information rich data.

### 5.2.3 Techniques for soft sensing

There exists a wide variety of soft sensor techniques as Figure 5.9 illustrates. Among those, the more representative examples are moving average model (Grazia ni et al., 2008), principal components analysis (PCA) (Dunia and Qin, 1998; Warne et al., 2004; Wang and Xiao, 2004; Lin et al., 2005), non-linear PCA (Wang et al., 2014), partial least squares (PLS) (Lin et al., 2007; Zhang and Lennox, 2004), dynamic PLS (Shang et al., 2015), support vector machines (SVM) (Kaneko and Funatsu, 2013), artificial neural networks (ANN) (Shang et al., 2014; Graziani et al., 2010; Fortuna et al., 2009, 2005a, 2007a; Liu et al., 2013; Jianxu and Huihe, 2002; Masson et al., 1999; Rogina et al., 2011; Rallo et al., 2003; Alhoniemi et al., 1999), subspace identification (Chokshi, 2012; Kano et al., 2009), and genetic algorithms (Mendes et al., 2012). Hybrid approaches include: PCA and radial basis functions ANN (Salahshoor et al., 2009; Yu et al., 2006), kernel least squares and SVM (Li et al., 2012), PCA and Gaussian process regression (Ge et al., 2011), PCA and ANN (Linhares, 2010; Rebouças, 2009), least squares and SVM (Gomnam and Jazayeri-rad, 2013), independent component analysis and PLS (Kaneko et al., 2008), and kernel PCA and SVM (Yang and Huang, 2010).

### 5.2.4 Special techniques for soft sensing

In addition to the abovementioned methods, other highly evolved techniques derived from machine learning and data mining fields were also applied in order to improve the system identification and, consequently, the development process. These include the ensemble methods, the local-learning, and the meta-learning (see Figure 5.9).

#### ■ Ensemble methods

The ensemble methods generate a set of models (also called ensemble members) and aggregate them to predict a system variable.

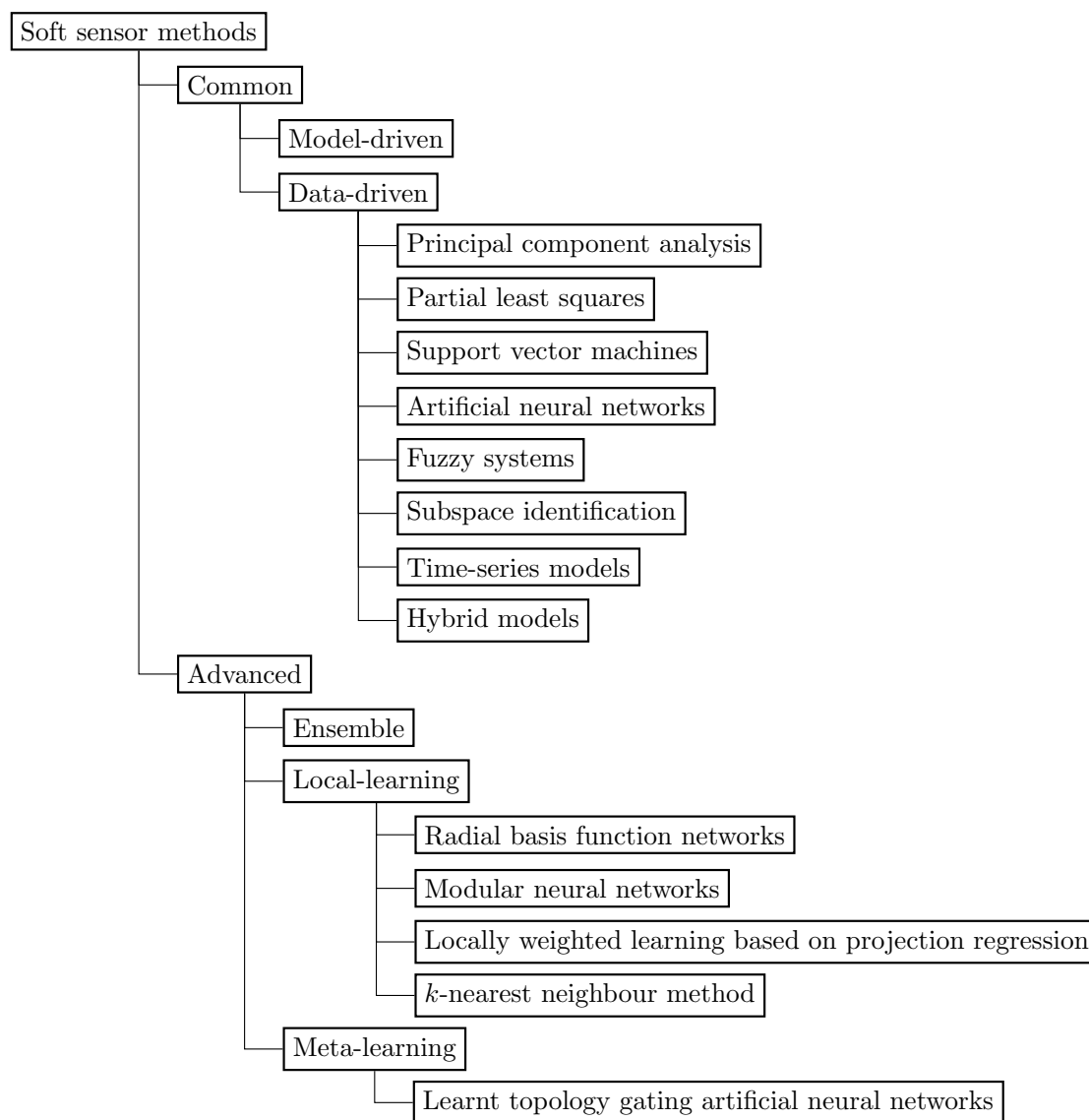


Figure 5.9: Methods for soft sensing.

The most common approaches to generate the set of models are the bagging method, the boosting method, and the modular neural networks. The bagging method (Breiman, 1996) generates multiple training sets of constant size by re-sampling the training data with replacement that are then used to train each ensemble member. The boosting method (Schapire, 1990) consists in an iterative training, where the training set for each new ensemble member is drawn from the points misclassified (or poorly predicted) by the previous member. Finally, the modular neural networks (Jacobs et al., 1991) specializes the ensemble mem-



bers on regions of the input space by a gating network. All these approaches generate a set of  $p$  ensemble members  $\mathcal{F} = \{f_i\}_{i=1}^p$ .

The aggregation of the ensemble members is the second most important operation in ensemble methods. Several combination strategies are studied in literature (Soares et al., 2011). Four common strategies are

- the best model selection via

$$f = \underset{f_i}{\operatorname{argmin}} g\left(f_i(\mathbf{x}_{\text{val}}, \mathbf{y}_{\text{val}})\right), \quad (5.18)$$

where  $g(\cdot)$  is an error function, and  $(\mathbf{x}_{\text{val}}, \mathbf{y}_{\text{val}})$  are the input and output variables of the validation set;

- the simple mean

$$f = \frac{1}{p} \sum_{i=1}^p f_i, \quad (5.19)$$

used with the bagging method;

- the trimmed mean with the removal of some models  $f_i$  before applying the mean in (5.19);
- the weighted average based on the accuracy

$$f = \sum_{i=1}^p w_i f_i, \quad (5.20)$$

$$w_i = \frac{C_i}{\sum_{i=1}^p C_i}, \quad (5.21)$$

where  $w_i$  are the combination weights with  $\sum_{i=1}^p w_i = 1$ ,  $\forall i : w_i > 0$  and  $C$  represents the correlation coefficient between the estimated and the real outputs.

However, Kadlec (2009) refers that the best model selection strategy is not efficient. Moreover, the quadratic error of the ensemble in the simple mean strategy decreases with the increasing of the ensemble size. Furthermore, tests performed

by Soares et al. (2011) showed that the simple mean, the trimmed mean and the weighted average based on the accuracy have better performances.

Some soft sensor studies based on ensemble methods may be found in Kordon et al. (2004), Jordaan et al. (2004), Jordaan et al. (2006), Minku et al. (2010), Soares et al. (2011), Liu et al. (2012), ZHANG Wenqing (2012) and Kaneko and Funatsu (2014). Kordon et al. (2004), Jordaan et al. (2004), and Jordaan et al. (2006) developed soft sensors based on genetic programming and the ensemble method while Kaneko and Funatsu (2014) used online support vector regression and a Bayesian-based ensemble method. Soares et al. (2011) studied several combinations of the ensemble methods in diverse conditions (such as the noise injection) proving their success in the chemical oxygen demand estimation in a pulp process. Finally, the impact of concept drift on the diversity of on-line ensemble methods was studied in Minku et al. (2010).

### ■ Local-learning

Evolved from the lazy learning, local-learning (Bottou and Vapnik, 1992; Atkeson et al., 1997) is a supervised learning that trains a set of models on limited partitions of the data space. In these methods, all the samples are collected from the plant and memorized in the database. When calculating a prediction for a given iteration (called query), the method searches in the database those samples that are contained in the neighbourhood of the query and builds a new model based on them.

The evaluation criterion, the kernel function, and the local model are three essential components in a local learning method. The first component evaluates the database and selects the neighbourhood samples of a given query  $\mathbf{q}$ .  $k$ -nearest neighbours algorithm based on the distance function may be used to select the neighbourhood. Commonly, the Euclidean distance is used as the evaluation criterion and is calculated by

$$d(\mathbf{x}, \mathbf{q}) = \frac{1}{m} \sqrt{(\mathbf{x} - \mathbf{q})(\mathbf{x} - \mathbf{q})^T}, \quad (5.22)$$

where  $\mathbf{x}$  refers to the selected neighbourhood samples (training set) and  $m$  to the number of samples.

The kernel function describes the form of the local neighbourhood and is commonly an expression of the exponential functions family. The kernel size is an important parameter of the kernel function that has a crucial influence on the locality of the model. It may be considered constant or variable with the density of the data samples.

Finally, the local model trains the neighbourhood data previously selected by the kernel function to calculate the prediction value. Several techniques may be used, including linear and non-linear regression (see Chapter 2).

The choice of these three components is a trade-off between locality and capacity. While locality defines the size of the neighbourhood, capacity defines the complexity of the local method. This is an important advantage of the local-learning method, because its performance (or its capacity) may be defined and controlled through the size of the training dataset (or through the locality). Local-learning methods in soft sensors development are also promising due to the ability to recognize different operating states of the process (through the data clustering). Consequently, the most adequate training method will be used instead of the training of a global model with the complete dataset.

Some applications of local methods that cope with drifts in process characteristics as well as non-linearity are presented in Zheng and Kimura (2001); Zeng et al. (2011) and Fujiwara et al. (2009). Economical impact of soft sensors based on this learning and their challenges are depicted in Kim et al. (2013). A method to avoid over-fitting in local learning is also proposed in Shao et al. (2013). In addition to the referred methods, local learning methods also include the following:

**Radial basis function networks:** Usually composed by three layers (the input, the hidden, and the output layers), the radial basis function networks (RBFN) (Orr, 1996) are distinct from other types of artificial neural networks because of the activation function used in the hidden layer and of the way that these layer weights are calculated. The neurons of the hidden and output layers are characterized by a Gaussian and a linear activation functions,

respectively, defined as

$$z_i^{\text{hidden layer}} = \exp \left( -\frac{1}{2} (\mathbf{x}^{\text{in}} - \mu_i)^\top \Sigma_i (\mathbf{x}^{\text{in}} - \mu_i) \right), \quad (5.23)$$

$$y_i^{\text{output layer}} = \mathbf{w}_i \mathbf{z}^{\text{hidden layer}}, \quad (5.24)$$

where  $z_i^{\text{hidden layer}}$  is the neuron  $i$  output of the hidden layer,  $\mathbf{x}^{\text{in}} \in \mathbb{R}^{n \times m}$  is the input space,  $\mu$  is the mean values vector defining the function centre in the input space,  $\Sigma$  is the covariance matrix defining the function spread in the input space,  $y_i^{\text{output layer}}$  is the neuron  $i$  output of the output layer, and  $\mathbf{w}_i$  is the weight vector associated to the output neuron  $i$ .

From the equations analysis, RBFN parameters are the vector  $\mu$  and the matrices  $\Sigma$  and  $\mathbf{w}$ . The parameters related to the Gaussian activation function,  $\mu$  and  $\Sigma$ , are calculated off-line by clustering the training data into  $N$  clusters (where  $N$  corresponds to the number of neurons in the hidden layer) and by calculating the respective function parameters for each cluster. Additionally, the weights  $\mathbf{w}$  are calculated by applying the least squares

$$\mathbf{w} = \left( (\mathbf{z}^{\text{hidden layer}})^\top \mathbf{z}^{\text{hidden layer}} \right)^{-1} \mathbf{z}^{\text{hidden layer}} \mathbf{y}^{\text{output layer}}. \quad (5.25)$$

From the view point of local learning, the construction of the hidden layer may be seen as the instances selection process while the output layer is seen as the local model building the final prediction.

**Modular neural networks:** Also classified as an ensemble method, modular neural networks (Jacobs et al., 1991) may be considered as a local-learning method. In a first step,  $p$  ensembles (called local experts) are trained using subsets of the training data. Then, the outputs of these networks  $\hat{y}_i$  are used to build new networks (called gating networks) in order to obtain the  $p$  local experts weights  $w_i$  that will decide the final output prediction. These weights are obtained by the gradient descent technique where the global error function

$$J = -\log \left[ \sum_{i=1}^p w_i \cdot \exp \left( -\frac{1}{2} \|y - \hat{y}_i\|^2 \right) \right] \quad (5.26)$$

is minimized enforcing the local experts specialization. In (5.26),  $y$  is the target value.

**Locally weighted learning based on projection regression:** It is a method that combines the advantages of the locally weighting learning (the speed, the efficiency, and the incremental capabilities) with the dimension reduction techniques (the capabilities to deal with a large number of inputs, possibly collinear) to approximate non-linear behaviours (Vijayakumar et al., 2005). Given a prediction request for  $\mathbf{x}_k$ , all the  $p$  local models  $f_i(\mathbf{x}_k)$  calculate their predictions  $\hat{y}_i$  and the total output prediction  $\hat{y}$  is obtained by

$$\hat{y} = \frac{\sum_{i=1}^p w_i(\mathbf{x}_k) \cdot \hat{y}_i}{\sum_{i=1}^p w_i(\mathbf{x}_k)}. \quad (5.27)$$

Here, weights  $w_i$  are determined using the Gaussian kernel defined by

$$w_i(\mathbf{x}_k) = \exp\left(-\frac{1}{2}(\mathbf{x}_k - \mathbf{c}_i)^\top \mathbf{D}_i(\mathbf{x}_k - \mathbf{c}_i)\right), \quad (5.28)$$

where  $\mathbf{c}_i$  is the central point of the validity region for the local model  $f_i$  and  $\mathbf{D}_i$  is a positive semi-definite distance metric determining the size and shape of the neighbourhood.

The neighbourhood selected by (5.28) is pre-processed to ensure zero mean for all inputs and outputs as well as uncorrelated inputs by PLS method application (see Section 2.2.4). Finally, the method finds a local approximation using the standard linear regression model.

**$k$ -nearest neighbour method:** The  $k$ -nearest neighbour or kNN method (Härdle, 1990) is among the simplest machine learning methods and uses weights  $w_i$  to set more significant contributions to the closer neighbours  $(x_{\text{neigh}}, y_{\text{neigh}})$  by

$$\hat{y}(x) = \frac{1}{m} \sum_{i=1}^m w_i(x_{\text{neigh},i}) y_{\text{neigh},i}. \quad (5.29)$$

Given a new query point  $x$ , kNN estimates the outcome based on the existing dataset. For regression problems, KNN predictions are based on averag-

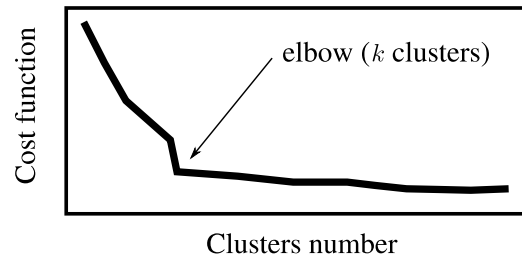


Figure 5.10: Representation of the elbow in the cost function.

ing the outcomes of the found  $k$  nearest neighbours of the query point. The Euclidean distance is usually taken to define the neighbourhood measuring the distance between the query point and the samples in the dataset.

The optimal clusters number  $k$  in kNN is an essential parameter often chosen based on experience or knowledge about the problem. The elbow method is a method used to help the determination of  $k$  that searches for an elbow in the curve cost function  $J$  (usually the mean square error or MSE) versus clusters number using the following algorithm:

1. vary the clusters number and compute the corresponding cost function;
2. as the clusters number increase, the cost function should decrease (if  $J = \text{MSE}$ );
3. plot the cost function versus the clusters number;
4. find the elbow point of the curve;
5. identify the clusters number  $k$  of the elbow point.

Unfortunately, the elbow identification is often performed by the user because usually the plot does not clearly show the elbow.

### ■ Meta-learning

Meta-learning methods (Vilalta et al., 2010) are supervised learning methods applied on meta-data (such as data assumptions, problem properties, and performance measures) from learning experiments trying to understand how to improve the prediction performance. Each expert performs well only when the application assumptions are similar to those they are based-on. Otherwise, they

become non-experts and probably will fail the prediction, which limits their applicability. The usage of a meta-learning method may select, alter or combine different learning algorithms in order to solve this problem in an effective and automatic way.

The meta-learning methods usually intend to find an hypothesis  $h$  from the hypothesis space  $H_L$  for a given learning algorithm  $L$  and using the meta-data space  $S^{train}$  to improve the prediction performance. Therefore, an objective function (usually called bias) is defined considering the most important aspects of the target hypothesis. These aspects may be: the algorithm parameters (e.g., number of layers and neurons in the ANN algorithm), the algorithm initialization (e.g., initial weights of the ANN algorithm), and the data pre-processing.

### Learnt topology gating artificial neural networks:

Learnt topology gating artificial neural networks (Kadlec and Gabrys, 2008b) applies meta-learning to an ensemble model based on modular neural networks. The approach generates ensemble members and weights them in order to predict the output variable. Considering Figure 5.11, the generation of the ensemble members based on modular neural networks is performed in the following way. In a first phase,  $p$  artificial neural networks

$$\hat{y}_{ji} = f_j(\mathbf{x}_i) \quad (5.30)$$

with a random number of hidden units are trained using  $N$  training data points  $(\mathbf{x}_i, y_i)$ . They are called base models or local experts (LE). Then, new ANN

$$\hat{w}_{ji} = g_i(\mathbf{x}_i) \quad (5.31)$$

are trained in order to obtain the prediction performance  $\hat{w}_{ji}$  for each expert  $j$  and data point  $\mathbf{x}_i$ . The observed prediction performance  $w_{ji}$  is calculated by

$$w_{ji} = \frac{1}{1 + (\hat{y}_{ji} - y_i)^2}. \quad (5.32)$$

These models are called gating ANN (GANN). Combining the two ANN

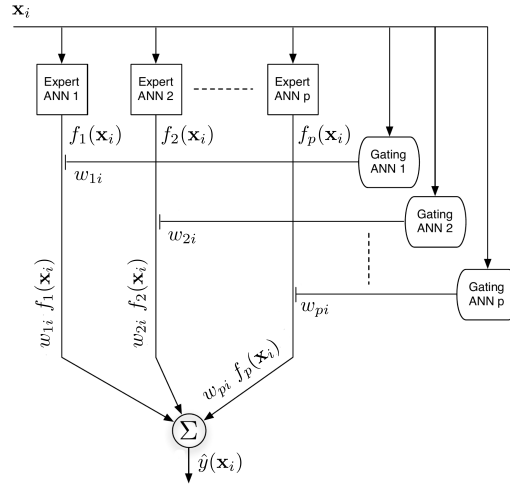


Figure 5.11: Learned topology gating artificial neural network.

types, the final prediction for a given  $\mathbf{x}_i$  is calculated as

$$\hat{y}(\mathbf{x}_i) = \sum_{j=1}^p \hat{w}_{ji}(\mathbf{x}_i) \hat{y}_{ji}(\mathbf{x}_i). \quad (5.33)$$

Restricting to three layer networks (input, hidden and output layers), the meta-learning approach in Kadlec and Gabrys (2008b) considers an initial distribution of the number of hidden units from equal distributions  $\mathcal{U}(H_{\text{LE}})$  and  $\mathcal{U}(H_{\text{GANN}})$  for the local experts and gating ANNs, where  $H$  represents the range of the hidden units number. After the ANNs generation and the evaluation of the LE and GANN performance (performance indices  $q_{\text{LE}}$  and  $q_{\text{GANN}}$ ), the distributions  $\mathcal{U}(\cdot)$  are modified towards the conditional probability distributions

$$\mathcal{U}^*(H_{\text{LE}}) \sim p(\mathcal{U}(H_{\text{LE}})|q_{\text{LE}}), \quad (5.34)$$

$$\mathcal{U}^*(H_{\text{GANN}}) \sim p(\mathcal{U}(H_{\text{GANN}})|q_{\text{GANN}}). \quad (5.35)$$

The modification will result in the addition and removal of new local experts with a new neurons number in the hidden layer. This meta-learning approach may deal with a fundamental ANN approach drawback which consists of the manual definition of the neurons number.



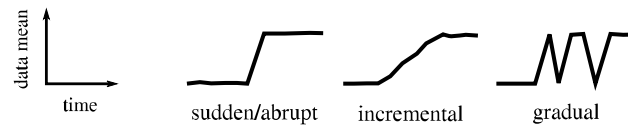


Figure 5.12: Concept drift patterns that may occur over time.

### 5.2.5 Detection and handling of concept drift

Concept drift refers to changes over time in the conditional distribution of the output variables given the input variables in some hidden context, while the distribution of the inputs may stay unchanged (Gama et al., 2014). Concept drift has important consequences because most of the learning procedures are based on data distribution and, when it does not hold, their performance becomes unreliable. Therefore, detecting concept drifts is of vital importance for applications working in dynamical environments, such as soft sensors (Dries and Rückert, 2009). It is important not to mix a true drift with outliers, noise, process anomalies or one-off random deviation (Gama et al., 2014). Also, the ability to adapt to a concept drift is a very important feature to guarantee an accurate performance of the application.

Usually, they are distinguished into virtual and real concept drifts. The former is a change in the distribution of the input data while the later is an actual change of the concept drift or, in other words, in the relation between the input and output variables. Concept drifts may also be distinguished by the form they manifest over time (Figure 5.12). Drifts may happen suddenly or abruptly (e.g. replacement of a sensor by another sensor with different calibration), incrementally (e.g. sensor slowly wears off and become less accurate), and gradually. Minku et al. (2010) and Kosina et al. (2010) characterized drifts by their severity, predictability, and frequency.

Over the last years, learning in the presence of concept drifts has been the subject of several works (Tsymbal, 2004; Kuncheva, 2004, 2008; Gama et al., 2014; Moreno-Torres et al., 2012) with a focus on concept drift detection and handling as described below.

### ■ Detection methods

Handling or adaptation methods may develop a new model at regular intervals without checking if a drift occurred or only when a drift is detected. In the later approaches, the detection is performed by monitoring some indicators over time. For classification problems, several detection methods were developed: the drift detection method (Gama et al., 2004), the early drift detection method (Baena-García et al., 2006), and the Hoeffding trees (Domingos and Hulten, 2000). For regression problems, the existing methods are scarce. Kadlec and Gabrys (2009) (Section 4.1.1) developed a detection method based on the residual vectors between the eventual training data  $\mathbf{y}$  and the model prediction  $f^{lm}(\mathbf{x})$  defined by

$$\mathbf{r} = \mathbf{y} - f^{lm}(\mathbf{x}). \quad (5.36)$$

The authors refer to the model  $f^{lm}(\cdot)$  as “landmarker”. Provided historical data, the first step of the algorithm is training a model using  $n_{init}$  samples from an initial window  $\mathcal{D}^{init}$  ( $n_{init}$  is a tuning parameter). Then, the window is shifted one step forward ( $s = 1$ ) while keeping the size  $n_{init}$  constant in the following form

$$\mathcal{D}^{shifted} = (\mathbf{x}^{shifted} - \mathbf{y}^{shifted}). \quad (5.37)$$

Following this, the residuals of the initial and the shifted sets are calculated and tested for a statistically significant difference using the  $t$ -test that considers the residuals as normally distributed. The aim is looking for a significant difference between the mean values of the two residuals vectors and, consequently, identify a significant change in the model performance as an effect of the concept drift. The procedure is repeated as long as the null hypothesis of the  $t$ -test remains valid.

Metrics commonly used to monitor the industrial processes performance may also be applied in the performance monitoring of soft sensors. The most used performance metrics are the Hotelling’s  $T^2$  and the squared prediction error (SPE) (Kadlec et al., 2011). The Hotelling’s  $T^2$  is calculated by

$$T^2 = \boldsymbol{\tau}^\top \boldsymbol{\Lambda} \boldsymbol{\tau}, \quad (5.38)$$

where  $\lambda = \text{diag}(\lambda_1, \dots, \lambda_l)$  is the weighting factors matrix and  $\Lambda$  are the latent vectors (for PLS model) or the principal components (for PCA model). The values of SPE for the output data  $y$  are obtained by

$$\text{SPE} = \|\mathbf{y} - \hat{\mathbf{y}}\|_2^2. \quad (5.39)$$

Considering that the Hotelling's  $T^2$  follows the  $F$ -distribution and SPE the central  $\chi^2$ -distribution, the calculation of these two metrics is followed by the computation of the respective confidence limits  $T_\beta^2$  and  $\text{SPE}_\beta$  by

$$T_\beta^2 = \frac{l(n^2 - 1)}{n(n - l)} F_{l, n-l, \beta}, \quad (5.40)$$

$$\text{SPE}_\beta = \theta_1 \left( 1 + \frac{c_\alpha \sqrt{2\theta_2 h_0^2}}{\theta_1} + \frac{\theta_2 h_0 (h_0 - 1)}{\theta_1^2} \right)^{1/h_0}, \quad (5.41)$$

where  $l$ ,  $n$ ,  $\beta$ , and  $c_\alpha$  are distribution parameters, and  $h_0$  and  $\theta_i$  (with  $i = 1, 2, 3$ ) are defined as

$$h_0 = 1 - \frac{2\theta_1\theta_3}{3\theta_2^2}, \quad (5.42)$$

$$\theta_i = \sum_{j=l+1}^m \lambda_j^i. \quad (5.43)$$

Finally, using the block-wise moving window model adaptation method (described below), the monitoring metrics may additionally be updated using

$$T_t^2 = \lambda T_{t-1}^2 + (1 - \lambda) T_{t-2}^2, \quad (5.44)$$

$$\text{SPE}_t = \lambda \text{SPE}_{t-1} + (1 - \lambda) \text{SPE}_{t-2}. \quad (5.45)$$

## ■ Handling methods

The most popular concept drift handling methods are based on the model adaptation and may be divided into three categories: instance selection methods, instance weighting methods, and ensemble methods (Figure 5.13). Although the performance monitoring is a very important issue in soft sensor application, it is commonly omitted and the soft sensor is continuously adapted without checking if the performance is actually degrading. It is the case of the categories instance

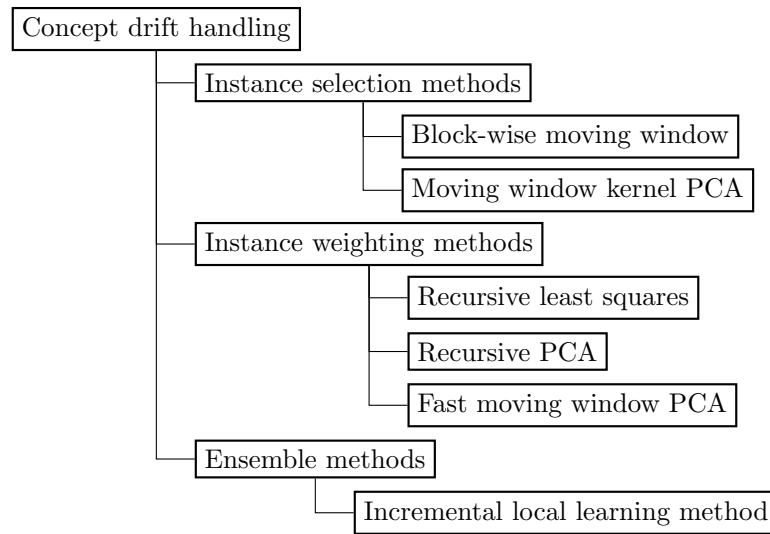


Figure 5.13: Concept drift handling methods.

selection and instance weighting methods (Kadlec and Gabrys, 2010). On the contrary, ensemble methods check the performance before the development of a new soft sensor using concept drift detection.

**Instance selection methods:** The main goal of the instance selection methods (also called moving window based methods) is to select relevant instances (samples) to the concept. The most common handling methods are based on this category and use moving window based techniques in which the model is updated or retrained using a set of samples recently arrived (Tsymbol, 2004). The set is selected in order to maximize the relevance for the current concept and, in the vast majority of cases, a fixed number of the most recent data points is assumed to be the most relevant to the current concept. As new instances are acquired, the window slides along the data so the newest samples are included and the oldest are excluded (Kadlec et al., 2011).

However, moving window techniques have some drawbacks, such as the parameters tuning and the data storage requirements. The tuning parameters associated with these techniques are the size of the adaptation window (usually referred to as window size) and the adaptation intervals between updates (usually designated by step size). An inappropriate tuning may lead to the performance degradation. Some approaches use adaptive mov-

ing windowing where the window size is adapted with the arrival of drifts (He and Yang, 2008; Kuncheva and Žliobaitė, 2009).

The adaptation by *block-wise moving window* is performed by retraining the model periodically after a given number of collected samples specified by the user as

$$f_t = \mathcal{L}(f^{init}, \mathcal{D}), \quad (5.46)$$

where  $f_t$  is the new soft sensor,  $f^{init}$  is the model structure,  $\mathcal{L}$  is the training algorithm,  $\mathcal{D}$  are the latest samples for the training composed by input and output variables.

Using the moving window method, Liu et al. (2009) uses the kernel PCA model to handle with non-linear relationships between variables creating the *moving window kernel PCA* method. In each window move which excludes the oldest sample  $\mathbf{x}_{k-n}$  and includes the newly available sample  $\mathbf{x}_k$ , the mean vector  $\mathbf{b}_k$  and the covariance matrix  $C_k$  are updated by

$$\mathbf{b}_k = \mathbf{b}_{k-1} + \frac{1}{n-1} \left( \phi(\mathbf{x}_k) - \phi(\mathbf{x}_{k-n}) \right), \quad (5.47)$$

$$\begin{aligned} C_k = C_{k-1} &- \frac{n-1}{(n-2)^2} \left( \phi(\mathbf{x}_{k-n}) - \mathbf{b}_{k-1} \right) \left( \phi(\mathbf{x}_{k-n}) - \mathbf{b}_{k-1} \right)^\top + \\ &+ \frac{1}{n-1} \left( \phi(\mathbf{x}_k) - \frac{n-1}{n-2} \mathbf{b}_{k-1} + \frac{k-n}{n-2} \phi(\mathbf{x}_{k-n}) \right) \cdot \\ &\cdot \left( \phi(\mathbf{x}_k) - \frac{n-1}{n-2} \mathbf{b}_{k-1} - \frac{k-n}{n-2} \phi(\mathbf{x}_{k-n}) \right)^\top, \end{aligned} \quad (5.48)$$

where  $\phi(\cdot)$  is a non-linear kernel function and  $n$  is the window length. Then, the kernel PCA model is recalculated using a numerically more efficient decomposition of the new covariance matrix  $C_k$  suggested by the authors.

In terms of data storage, both techniques require the storage in memory of all samples within the window which may be problematic for large windows.

**Instance weighting methods:** Instance weighting methods (also called recursive adaptation methods) use the current model and the new information to update the model. Usually, this is achieved by down-weighting the current model via the use of a forgetting factor. Because the weights are as-

signed to the samples according to their age, there is the need to choose the speed of the temporal decay of the sample weights (usually by tuning). And, similarly to moving window techniques, instance weighting methods do not provide any mechanism for concept drift detection needing information about window size and step size. The forgetting factor may eventually be adapted (Fortescue et al., 1981; Choi et al., 2006).

This approach has similar problems as the moving window approach. In addition, the instance weighting techniques adapt the models excessively when the process is operated within a narrow range for a certain period of time. Therefore, the methods will not cope with abrupt changes in the process because they will not function in a sufficiently wide range of operating conditions (Fujiwara et al., 2009).

The *recursive least squares* method is an offline version of the algorithm of least squares used incrementally to incorporate new incoming samples (Jang et al., 1997; Ljung, 1999). The least squares estimator

$$\theta = (X^T X)^{-1} (X^T Y), \quad (5.49)$$

where  $\theta$  are the linear regression coefficients and  $X$  and  $Y$  the input and output data matrices, may be rearranged in such a way that incorporates the old data and the new sample  $\mathbf{x}_t$  by

$$\theta_t = \left( \begin{bmatrix} X_{t-1} \\ \mathbf{x}_t \end{bmatrix}^T \begin{bmatrix} X_{t-1} \\ \mathbf{x}_t \end{bmatrix} \right)^{-1} \left( \begin{bmatrix} X_{t-1} \\ \mathbf{x}_t \end{bmatrix}^T \begin{bmatrix} Y_{t-1} \\ \mathbf{y}_t \end{bmatrix} \right) \quad (5.50)$$

or recursively by

$$\theta_t = \theta_{t-1} + P_t \mathbf{x}_t (\mathbf{y}_t - \mathbf{x}_t^T \theta_{t-1}), \quad (5.51)$$

where  $P_t = (X_t^T X_t)^{-1}$  is the inverted covariance matrix and may also be calculated recursively by

$$P_t = P_{t-1} + \frac{P_{t-1} \mathbf{x}_t \mathbf{x}_t^T P_{t-1}}{1 + \mathbf{x}_t^T P_{t-1} \mathbf{x}_t}. \quad (5.52)$$

In order to avoid that all samples be considered equal, this equation is re-

duced to

$$P_t = \frac{1}{\lambda} \left( P_{t-1} + \frac{P_{t-1} \mathbf{x}_t \mathbf{x}_t^\top P_{t-1}}{1 + \mathbf{x}_t^\top P_{t-1} \mathbf{x}_t} \right), \quad (5.53)$$

where  $\lambda$  is the sample weighting factor.

The *recursive PCA* (Li et al., 2000) provides efficiently a new PCA model by the updated correlation matrix calculation from the previous rather than by using the old data. Consider a first data matrix  $\mathbf{X}_k^o \in \mathbb{R}^{k \times m}$  composed by  $m$  process variables in  $k$  time instants with mean and standard deviation  $\mathbf{b}_k$  and  $\Sigma_k = \text{diag}\{\sigma_k(1), \dots, \sigma_k(m)\}$ , respectively. The scaled data matrix  $\mathbf{X}_k = \Sigma_k^{-1}(\mathbf{X}_k^o - \mathbf{b}_k)$  is used to calculate the correlation matrix  $\mathbf{R}_k$  by

$$\mathbf{R}_k = \frac{1}{k-1} \mathbf{X}_k^\top \mathbf{X}_k. \quad (5.54)$$

When a new sample  $\mathbf{x}_{k+1}^o$  arrives, the extended data matrix  $\mathbf{X}_{k+1}^o = \begin{bmatrix} \mathbf{X}_{k+1}^o \\ \mathbf{x}_{k+1}^o \end{bmatrix}$  has a new mean and standard deviation calculated using

$$\mathbf{b}_{k+1} = \frac{k}{k+1} \mathbf{b}_k + \frac{1}{k+1} \mathbf{x}_{k+1}^o, \quad (5.55)$$

$$\Sigma_{k+1} = \text{diag}\{\sigma_{k+1}(1), \dots, \sigma_{k+1}(m)\}, \quad (5.56)$$

where

$$\sigma_{k+1}^2(i) = \frac{k-1}{k} \sigma_k^2(i) + \Delta \mathbf{b}_{k+1}^2(i) + \frac{(\mathbf{x}_{k+1}^o(i) - \mathbf{b}_{k+1}(i))^2}{k}, \quad (5.57)$$

with  $\Delta \mathbf{b}_{k+1} = \mathbf{b}_{k+1} - \mathbf{b}_k$ . The scaled new sample  $\mathbf{x}_{k+1} = \Sigma_{k+1}^{-1}(\mathbf{x}_{k+1}^o - \mathbf{b}_{k+1})$  is now used to update the correlation matrix given by

$$\begin{aligned} \mathbf{R}_{k+1} &= \frac{k-1}{k} \Sigma_{k+1}^{-1} \Sigma_k \mathbf{R}_k \Sigma_k \Sigma_{k+1}^{-1} + \\ &+ \Sigma_{k+1}^{-1} \Delta \mathbf{b}_{k+1} \mathbf{b}_{k+1}^\top \Sigma_{k+1}^{-1} + \frac{1}{k} \mathbf{x}_{k+1} \mathbf{x}_{k+1}^\top. \end{aligned} \quad (5.58)$$

Wang et al. (2005) proposed the *fast moving window PCA*. Conventional moving window PCA considers a moving window that discards the oldest sample and adds a new one to the window. Then, it calculates a new PCA model

from the data matrix obtained from this new window, what is not efficient because the number of data points must be sufficient to be representative of the current plant operation. Fast moving window PCA tries to overcome this deficiency by combining the recursive PCA to enhance the adaptation mechanism. Recursive PCA brings the capacity to adapt without the need to process all the data points inside the moving window. The new method just adapts the mean and standard deviation of the variables and recursively determines the new correlation matrix. In rigour, this method may fall into both the instance selection and the instance weighting methods, because it merges methods of these two categories.

**Ensemble methods:** Ensemble methods have in memory an ensemble of multiple models to make a combined prediction. For concept drift handling, the ensemble methods may be used in three different ways: by dynamic combination, by continuous update, and by structural update (Gama et al., 2014). In dynamic combination, the individual models are dynamically combined to handle the process changes by modifying the combination weights. In continuous update, the models are retrained off-line or on-line modes using new data. Finally, in structural update, new models are added and the worst models are removed.

The *incremental local learning method* (Kadlec and Gabrys, 2008a, 2011) is based on the weighted combination of the  $p$  local models predictions translated by (5.20). The adaptation is performed by adapting the local models and the weights, simultaneously. The local models update is implicitly performed by using the recursive strategy associated to the PLS modelling. While the weights calculated through a 2D map, which models the relative prediction accuracy of the local models, are recalculated by an updated second map multiplication.

### 5.3 Glycerine evaporator application

Biodiesel is a fuel produced by a chemical reaction between vegetable or animal oil and alcohol (Aransiola et al., 2014). Its production costs are usually high which motivates the search for options that lower these costs. Since biodiesel produc-



tion generates about 1kg of glycerine per 10 kg of oil as the main by-product, its utilization to defray the biodiesel production costs is very important in the promotion of the large-scale production of biodiesel (Manosak et al., 2011; Tan et al., 2013). After purification, glycerine is an important high-value and commercial chemical with several uses in the food, cosmetic and pharmaceutical industries as explained by Tan et al. (2013). The optimization of the glycerine purification process is usually overlooked due to other tasks related with the main process of biodiesel production. Consequently, this part of the plant commonly runs at a suboptimal performance.

The importance of monitoring systems has been increasing in industrial plants as they contribute decisively to the high product quality and to environmental constraint compliance (Salahshoor et al., 2011). According to Grazia ni et al. (2008), soft sensors have been widely applied for on-line plant monitoring since the mid 80's for estimating process variables (usually product quality) by using mathematical models as an attractive low cost alternative to the expensive hardware analyzers.

One of the biodiesel manufacturing process byproducts is the mixture of water and glycerine that, in order to be marketable, has to be purified to reach a water composition around the 10%(w/w) via an energy intensive process. Usually, quality control is performed in the laboratory with large time delays and some lack of accuracy. Therefore, a soft sensor for real-time prediction of the glycerine quality is a tool with high potential for the biodiesel production process monitoring and optimization. The aim of the current section is to develop a soft sensor for the glycerine quality prediction to avoid measurement delays, to monitor the process in real-time, and to enable quick control actions to be taken by the operators or by a controller.

#### 5.3.1 Glycerine concentration process

Glycerine is used in numerous applications and may be obtained from oil via the transesterification reaction that occurs in biodiesel production (Tan et al., 2013). Before the application in new products, glycerine must be recovered from a crude solution with an initial water composition of 80–90%(w/w). Essentially, it involves three steps: pre-treatment, concentration, and refining.

In the pre-treatment phase, the non-glycerine components, such as dispersed fat, fatty acid, organic non-glycerine matter, and salt, are largely removed by settling, centrifugation, and other separation techniques. Subsequently, in the glycerine concentration step the water content is reduced to 10–30%(*w/w*). This is an important step that brings the product to a stable condition for storage. Finally, the refining step is used to remove the remaining water in the concentrated glycerine by using a distillation unit under high vacuum at around 160 °C (IPS Engineering, 2014).

Refining crude glycerine is costly as Dow (2015) states, so biodiesel plants usually choose to perform only pre-treatment and concentration steps because they already ensure a higher commercial product value and a substantial decrease of the production costs. The glycerine recovery from the aqueous solution is an energy intensive step. Therefore, its optimization may result in tangible economic benefits.

Figure 5.14 illustrates the glycerine concentration process provided by IPS Engineering (IPS Engineering, 2014). It consists of two multiple-effect evaporators followed by an heat exchanger and a flash tank. Initially, the feed composed by pre-treated crude glycerine is pre-heated. Then, it passes by two effect evaporators used to increase the steam economy. Steam feeds the tubes side of the first effect where the pre-heated feed enters. The vapour produced from evaporation in this first effect is fed to the second effect to provide the heat to evaporate more water from the product. The more concentrated glycerine that leaves the first evaporator feeds the second. The vapour produced in this second effect is directed to the tubes side of the heat exchanger where the concentrated glycerine from the second effect enters the shell side. Finally, the glycerine goes to a flash tank at low pressure where further evaporation takes place. The flash tank liquid stream is the concentrated glycerine product.

### 5.3.2 Soft sensor development

The basic steps in the development of the soft sensor (data collection, data pre-processing, model selection, model training and model validation) are described in the following sections.

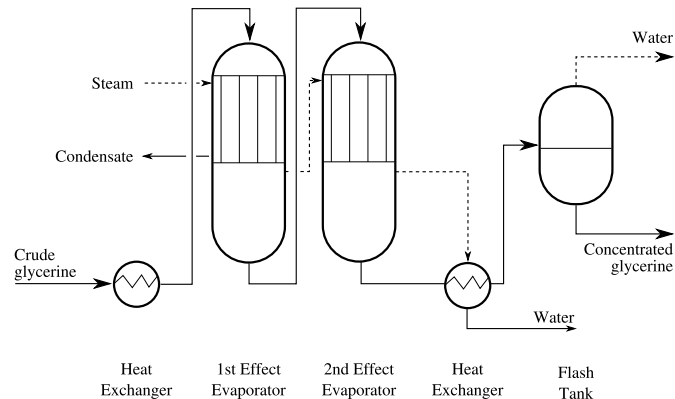


Figure 5.14: Diagram of the glycerine concentration process developed by IPS Engineering. Solid arrows represent liquid streams, while dashed arrows represent vapour streams.

### ■ Data collection

In order to predict the water composition in the concentrated glycerine using soft sensing technology, a dataset collected in industrial environment is used. It consists of six time series classified as input and output variables (Figure 5.15). The temperature in the first evaporator ( $x_1$ ), the temperature in the second evaporator ( $x_2$ ), the flash tank pressure ( $x_3$ ), the temperature of the vapour stream released in the flash tank top ( $x_4$ ), and the temperature of the liquid stream released in the flash tank bottom ( $x_5$ ) are the input variables. The water contents in the concentrated glycerine ( $y$ ) is the output variable. While input variables were collected from the plant historian with a sampling time of 1h, the output variable was measured in the plant laboratory by technicians. For confidentiality reasons, normalized data will be shown below.

### ■ Data pre-processing

The dataset was pre-treated in two steps. First, the data filling in was performed in the output variable because it consists of laboratory measurements with a variable sampling period. Hence, this dataset was filled in order to contain a sampling time of 1h assuming a linear model between two consecutive values. Second, outlier points were removed considering the mean and range of each variable to define the normal operational band. The obtained dataset contains

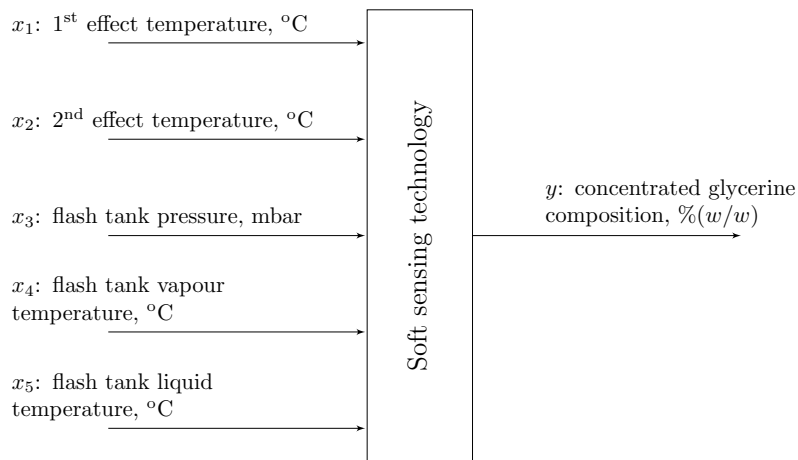


Figure 5.15: Soft sensor input and output variables.

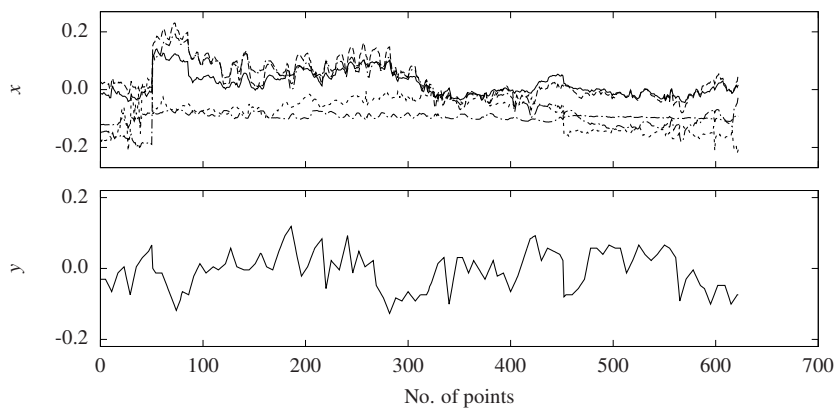


Figure 5.16: Selected data.

675 points and was normalized in order to avoid the overshadowing of variables with small magnitudes. Figure 5.16 shows the input and output variables after data pre-processing.

The data is divided into the training and validation subsets. The division is performed distributing alternately one data point for each subset.

### ■ Model selection

Considering the scheme presented in Figure 5.15, three modeling methodologies were applied: Partial Least Squares (PLS), Feedforward Artificial Neural Network (FANN), and Layer Recursive Artificial Neural Network (LRANN).

Developed by the Swedish statistician Herman Wold (Wold, 1982), PLS is a simple algorithm widely applied for relating variables by a linear multivariate model able to analyse noisy and collinear data (see Chapter 2). Most popular nonlinear methods found in the development of soft sensors are based on ANN because they are powerful tools for the modelling of complex multivariable processes (see Chapter 2). FANN is the simplest type of ANN where the connection between the inputs and outputs is performed in only one direction (forward). In contrast, LRANN contains connections between units in such a way that forms feedback loops creating a state that exhibits a dynamical behaviour (Schuster and Paliwal, 1997).

The PLS model was developed using all the components in the data, which means that all the capacity of the model was used to predict data ( $n_{co} = 5$ ). The ANN model development consisted of the model structure selection in the input, the hidden, and the output layers, composed by 5 (number of input variables),  $n_{hi} = 30$ , and 1 (number of output variables) neurons, respectively. In the LRANN model, a sixth input is added to introduce the time dependence. The number of past values introduced in this input is  $n_{td} = 3$ . For both ANN models, the aggregation function is the sum whereas the hyperbolic tangent and linear functions were chosen as activation functions for the hidden and output layers, respectively. The performance goal was defined by default as 0, while the maximum number of epochs was set to 200. ANN based models representations are shown in Figure 5.17.

The model training and validation was performed in Matlab R2014b programming language where several default functions such as `plsregress`, `net`, and `trainlm` were used.

### ■ Model training

Figure 5.18 shows the output variable prediction of the three models for the training set. In the left column, it is possible to compare the time trends of actual and predicted data. The plots in right column draw the predicted versus actual data.

To evaluate the models prediction capacity and the development performance, the mean square error (MSE), the coefficient of determination ( $R^2$ ), and the computational time of the model training ( $t_{comp}$ ) were determined for all of the tests

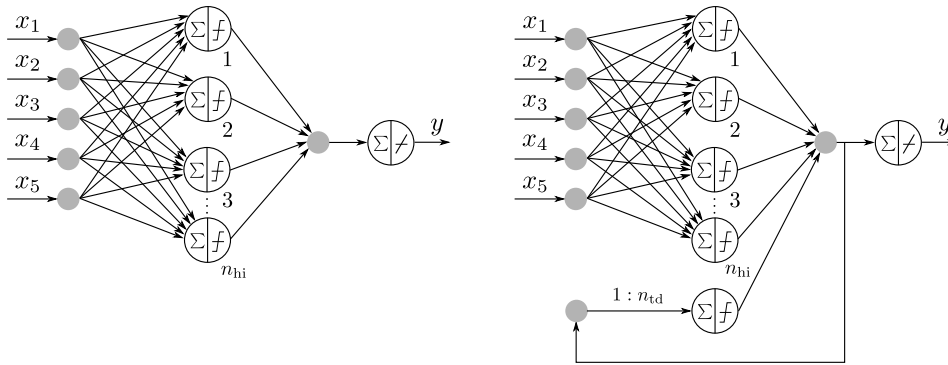


Figure 5.17: ANN based models representation: FANN (left) and LRANN (right).

and are in Table 5.1. An MSE value near zero and an  $R^2$  near one show a good prediction of the model. Values in rows c, f and i and columns 2 and 3 show the test results for the PLS, FANN, and LRANN models considered in the present section, respectively. The values of MSE and  $R^2$  obtained by the PLS model demonstrate its poorer predicting capability ( $MSE_c=0.00230$  and  $R_c^2 = 0.113$ ). Contrarily, the values of MSE ( $MSE_d=0.00038$  and  $MSE_h=0.00000$ ) and  $R^2$  ( $R_d^2 = 0.853$  and  $R_h^2 = 1.000$ ) of ANN based models show that they may predict quite well the system and that the LRANN approach rendered the best quality. Plots drawing actual versus predicted data confirm these conclusions. Since the first  $n_{td}$  past data points have no predicted values, it is assumed that the predicted variable has the same value as the measured variable in the first  $n_{td}$  time instances. Regarding the computational time, the training of the PLS is significantly less demanding than the LRANN.

### ■ Model validation

The model validation follows the training step and, as referred in Section 2.2.5, it should be performed using independent process data. Figure 5.19 shows both the process data and its estimation by the developed soft sensors. Performance measures are pointed out in rows c, f and i and columns 4 and 5 of Table 5.1. The visual inspection of the figures reveals that the ANN based models achieved better results than the PLS model. LRANN model is the best approach showing better performance values ( $MSE_g = 0.00042$  and  $R_g^2 = 0.838$ ) than the obtained with the FANN model ( $MSE_d = 0.00116$  and  $R_d^2 = 0.547$ ).

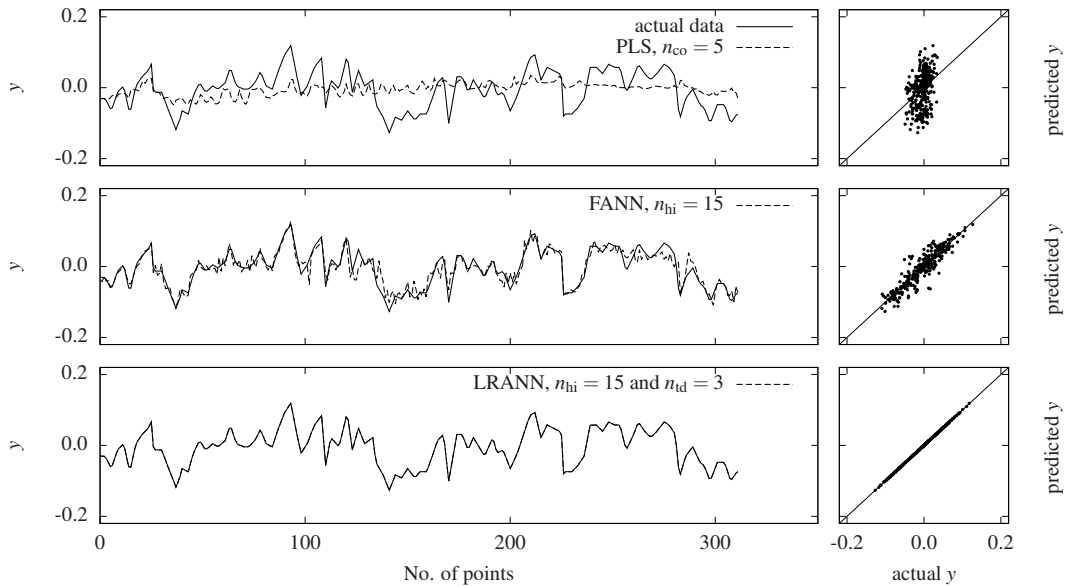


Figure 5.18: Results of the model training step.

### 5.3.3 Effect of model structure on prediction

In order to evaluate the model structure effect on the prediction of glycerine composition, a number of tests were performed varying the PLS model component number, the FANN and LRANN models hidden neuron number, and the LRANN model past point number and the results for the validation dataset are illustrated in Figures 5.20–5.23.

Figure 5.20 shows the effect of the PLS model components number. Three tests were performed using 3, 4 and 5 components. Using  $n_{hi} = 5$  components is equivalent to use all the process variables, while using fewer components explores whether the variability may be predicted using less plant data. Entries a, b and c of Table 5.1 show the performance measures for the training and validation datasets. Although the performance measures of the PLS model increases in the tests with more components, their values confirm that the prediction still is very weak for all the considered components. Consequently, the PLS model is not able to predict adequately the glycerine composition, possibly due to its inability of modelling nonlinear relationships.

Due to a possible data over-fitting observed in Section 5.3.2, the number of neurons that compose the hidden layer of the ANN based approaches was stud-

Table 5.1: Performance measures.

Test	Training set		Validation set		$t_{\text{comp}}, \text{s}$	
	MSE	$R^2$	MSE	$R^2$		
a	PLS, $n_{\text{co}} = 3$	0.00230	0.113	0.00228	0.113	0.0
b	PLS, $n_{\text{co}} = 4$	0.00229	0.116	0.00227	0.117	0.0
c	PLS, $n_{\text{co}} = 5$	0.00228	0.118	0.00226	0.119	0.0
d	FANN, $n_{\text{hi}} = 5$	0.00115	0.557	0.00127	0.506	2.2
e	FANN, $n_{\text{hi}} = 10$	0.00045	0.826	0.00092	0.642	1.3
f	FANN, $n_{\text{hi}} = 15$	0.00038	0.853	0.00116	0.547	1.3
g	LRANN, $n_{\text{hi}} = 5$ and $n_{\text{td}} = 3$	0.02004	0.000	0.02020	0.000	0.4
h	LRANN, $n_{\text{hi}} = 10$ and $n_{\text{td}} = 3$	0.00037	0.856	0.00240	0.066	5.3
i	LRANN, $n_{\text{hi}} = 15$ and $n_{\text{td}} = 3$	0.00000	1.000	0.00042	0.838	20.2
j	LRANN, $n_{\text{hi}} = 15$ and $n_{\text{td}} = 6$	0.00002	0.993	0.00201	0.215	138.2
k	LRANN, $n_{\text{hi}} = 15$ and $n_{\text{td}} = 9$	0.00008	0.971	0.00933	0.000	298.4

ied. Three tests using 5, 10 and 15 neurons in the hidden layer were performed. Figures 5.21 and 5.22 show the neurons number effect on the glycerine composition prediction. Table 5.1 lists the performance measures in rows d, e, and f (for the FANN model) and g, h, and i (for the LRANN model with a constant  $n_{\text{hi}} = 3$ ).

For the training dataset, FANN model performance is superior when a higher number of neurons is used, because it allows to predict the glycerine composition with higher accuracy. However,  $n_{\text{co}} = 10$  neurons are preferable due to lower validation error. Regarding the LRANN model, the fit quality in the training step is again superior when a higher number of neurons is used achieving the highest performance when  $n_{\text{co}} = 15$ . The lowest validation error was also achieved with the number of neurons in the hidden layer of the LRANN model equal to 15. The dynamic component of the LRANN model helps to improve the FANN model results (as shown by the higher values of  $R^2$  and lower values of MSE). This suggests that the system has a dynamic state.

The effect of the number of past data points that the recurrent connection considers was also studied. Three tests using 3, 6 and 9 past values re-introduced in the hidden layer were performed. Figure 5.23 presents the results for the validation dataset and the performance measures may be found in rows i, j and k of Table 5.1. An increase in the past data input leads to poorer prediction capability that suggests that the sampling time is higher than the time constant of the



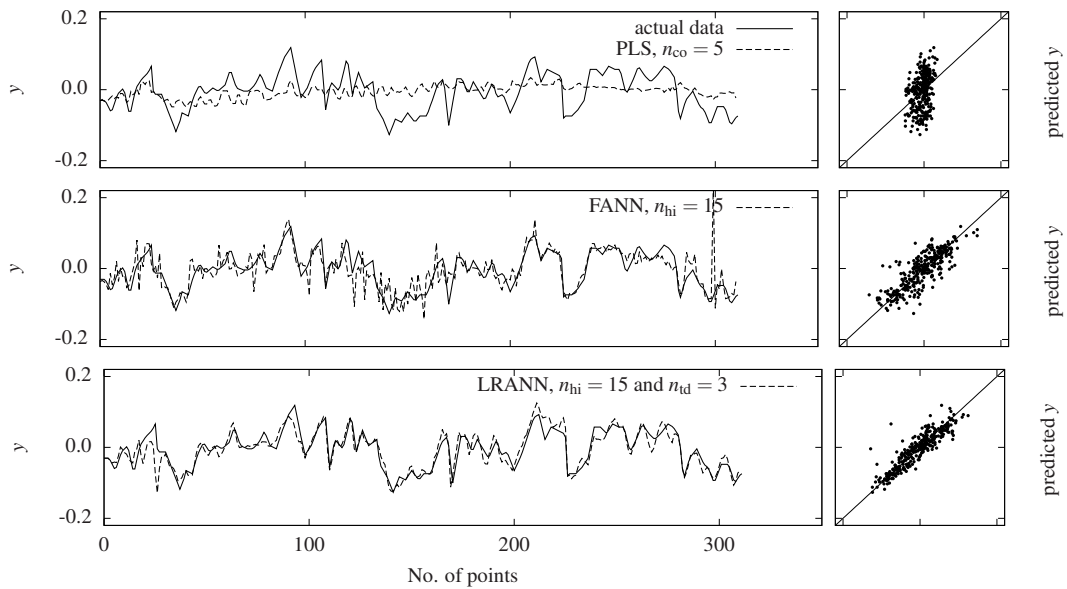


Figure 5.19: Results of the model validation step.

process.

When considering the validation dataset, the models with  $n_{td} = 6$  and  $n_{td} = 9$  failed the prediction, possibly due to over-fitting.

Comparing the results of the computational time used in the models training phase, it is possible to conclude that a LRANN model needs substantially more time to perform the training than the other approaches. In contrast, the PLS model is the fastest approach.

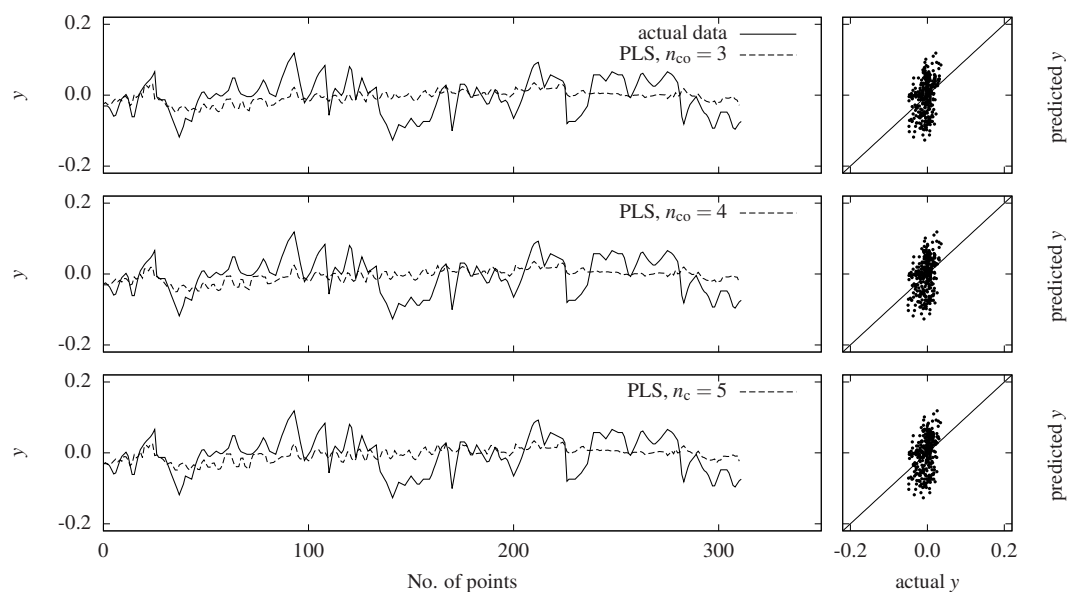


Figure 5.20: Effect of PLS model components number in the glycerine composition prediction.

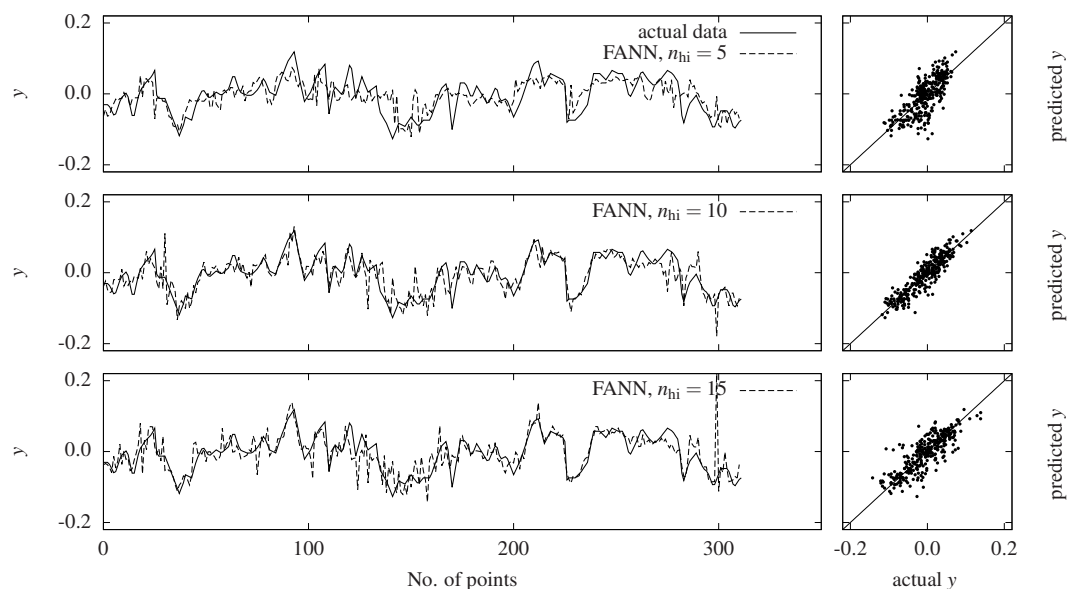


Figure 5.21: Effect of FANN model hidden layer neurons number in the glycerine composition prediction.

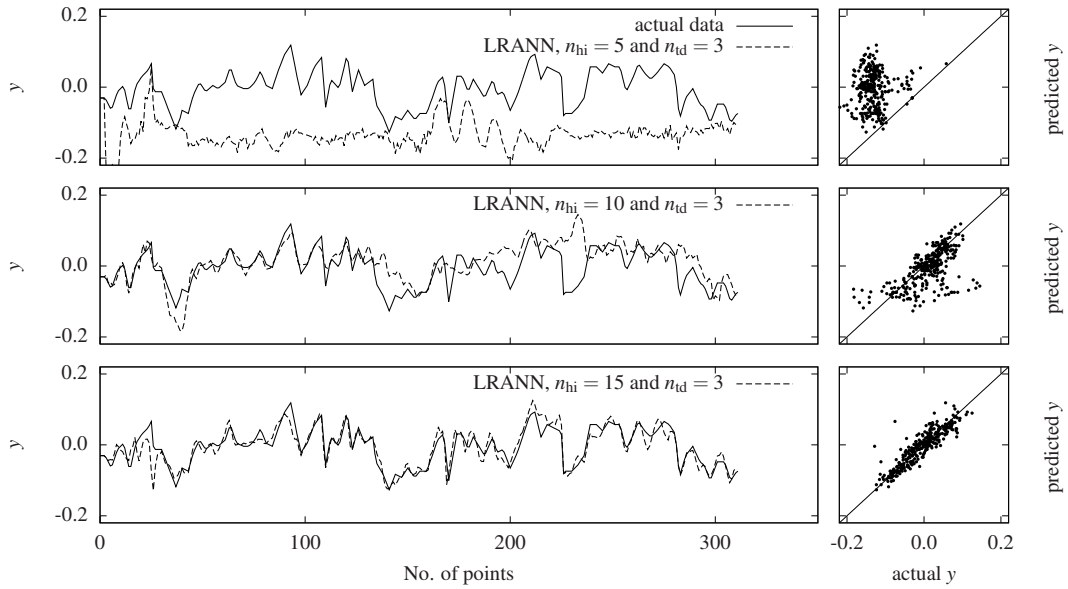


Figure 5.22: Effect of LRANN model hidden layer neurons number in the glycerine composition prediction.

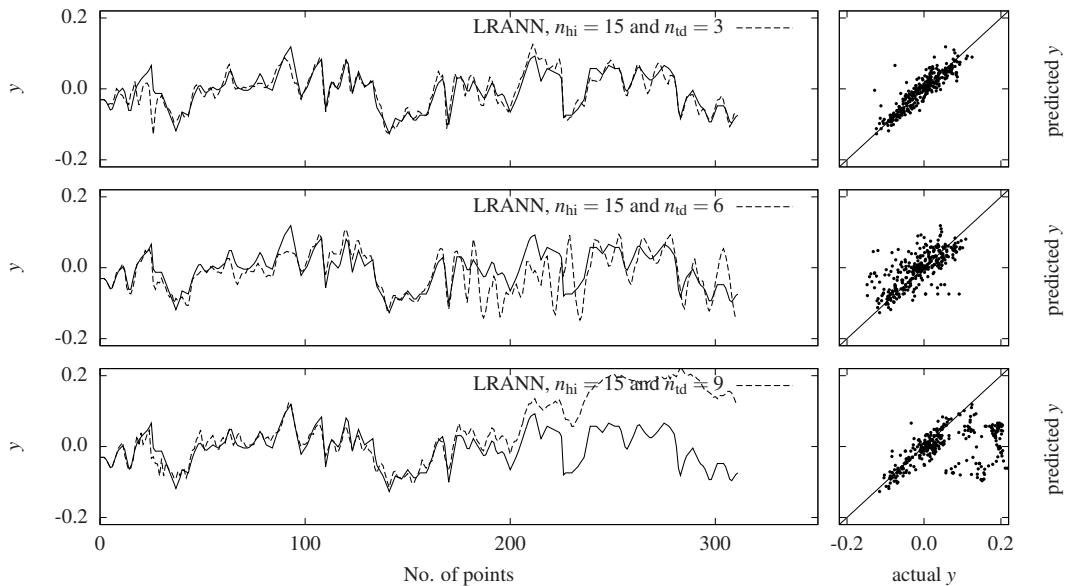


Figure 5.23: Effect of LRANN model time delay in the glycerine composition prediction.



# Chapter 6

## Conclusions and future work

In this chapter, the main contributions of the present research work in control loops performance monitoring and improvement are marked. Possible future research directions are also highlighted.

### 6.1 Main contributions

The research work covered in this thesis deals with performance monitoring and optimization of industrial control loops, which is critical in order to maximize the economic output of production assets subject to product quality specifications, operational, safety, and environmental constraints. Both the theoretical method development and the numerical implementation of the algorithms have been carried out.

#### 6.1.1 System Identification

Chapter 2 comprises a review of closed-loop system identification and two case studies based on two chemical engineering problems.

The first presented the development of a system identification approach and its computational implementation for SISO and MIMO systems. First- and second-order linear models in the state-space were obtained from industrial heat exchanger data. Since their performance revealed to be very similar, the first-order model was selected due to its simpler structure and lower computational effort. Both linear models were also applied to a simulated dataset of a CSTR equipped

with a heating coil and that has two input variables (the inlet flow concentration of reactant and the coil temperature) and two output variables (the reactant concentration and the reactor temperature). The parameter estimation of the first and second-order models showed some convergence difficulties and revealed to be sensitive to the initial estimate. These problems were successfully overcome by applying parameter normalization and by choosing a more adequate initial model estimate.

The second case study consists of the development of an hybrid dynamic model of a decanter of a biodiesel plant. The hybrid model is composed by a first principle model to describe the dynamics of the biodiesel decanter and by an artificial neural network model of the liquid-liquid equilibrium among glycerol, ester (the main compound of biodiesel), and methanol. Besides the complete study of the decanter dynamics, this work utilized an alternative to the commonly used iterative flash calculations with the purpose of reducing substantially the computation time of the model and enabling an online implementation.

### **6.1.2 Control Loop Performance Assessment**

Since Harris (1989) work, control loop performance monitoring methods have enjoyed a continuous interest in the academic and industrial world. Chapter 3 provides a state-of-the-art review and a systematic taxonomy of existing controller performance monitoring approaches. Special attention was given to stiction, a long-standing control valve problem, its modelling, detection/quantification, covering more than 150 publications.

The presence of stiction in control valves impacts the behaviour of the control loop and produces sustained oscillations. A significantly improved method of detection and characterization of multiple oscillations was proposed in Section 3.3. The approach is based on a single calculation of the auto-correlation function and was successfully applied to 3 cases where other approaches had failed, namely where the signals were affected by noise or multiple frequency oscillations as well as signals in which stiction was absent. Besides, the proposed method requires a light computational burden and may be run automatically.

Two new approaches for stiction detection were also proposed in this thesis (see Sections 3.4 and 3.5). The first approach detects and quantifies stiction using

numerical optimization. It uses the Hammerstein Model of the control valve affected by stiction and of the industrial process. The discontinuous valve stiction part of the model was smoothed via the hyperbolic function in order to enable the use of continuous optimization technique. The developed approach reproduced quite well the experimental data and was able to detect and quantify correctly valve stiction present in the control loops.

The second approach, based on pattern recognition, extended the stiction detection method of Yamashita (2006a) to integrating processes such as level control loops. The data is preprocessed by a transformation function to obtain a direct relation between the controlled variable and the valve position. This approach was successfully applied to simulated and industrial datasets from integrating processes. A study about the influence of the noise was also carried out. Although the stiction phenomena gets obfuscated by noise, correct stiction diagnosis is possible with data filtering. The method proved to outperform the existing ones, namely when dealing with noisy data.

### 6.1.3 PID controller tuning

In Chapter 4, a tuning method of PID controllers was developed that is capable of mitigating the impact of valve faults (such as stiction) until the maintenance work is possible. After selecting an appropriate dataset, the method checks for faults in the control valve. If some fault is identified, two mathematical models (one for the process and other for the fault) are identified. Otherwise, the nominal process model is obtained.

The determination of the controller tuning parameters is formulated as an optimization problem whose objective function penalizes both the deviation of the controlled variable from the setpoint and the valve movement. Besides, the user may specify additional performance criteria and variables constraints in order to obtain the desired closed-loop response.

A detailed application on a flow rate control loop was demonstrated using two simulated datasets that incorporate a healthy and a sticky valve. The method produced correct stiction diagnosis results in both cases. Several sets of tuning parameters were determined for the same open-loop process by varying the performance criteria and constraints. The behaviour of the control loop with a sticky

valve was significantly improved reducing the control moves and the oscillations in the controlled variable.

### 6.1.4 Soft sensor technology

In Chapter 5, soft sensor technology was addressed as a way to generate new information that is not readily available from on-line instrumentation or laboratory measurements, critical for real-time process monitoring and control.

The contribution in this thesis intended to broaden the use of soft sensors technology in industrial scenarios.

Although the glycerine concentration process is not the main production unit of a biodiesel plant, it is one of the most energy intensive. However, the use of on-line analyzers for glycerine quality prediction is not common and, therefore, soft sensor technology has a high potential for this task. In a case study, three soft sensors were developed and their performance in the prediction of glycerine quality was analyzed. The layer recursive ANN model showed the best prediction quality in both the training and validation phases due to its capacity of reproducing nonlinear relationships and dynamical behaviours.

### 6.1.5 Publications

This thesis has been the outcome of four years of research and development and a significant portion has been reviewed and published. An overview of these contributions is given below.

#### System Identification

- Brásio, A. S., Romanenko, A., and Fernandes, N. C. (2012). System identification as an application of optimization. In Simos, T. E., Psihoyios, G., Tsitouras, C., and Anastassi, Z., editors, *Proceedings of the AIP Conference*, volume 1479, pages 822–825, Kos (Greece). AIP. URL <http://dx.doi.org/10.1063/1.4756264>
- Brásio, A. S., Romanenko, A., and Fernandes, N. C. (2013). Aproximação de cálculos iterativos por redes neuronais em sistemas de equações diferenciais



ordinárias. In *Proceedings of the XVI Congress of the Portuguese Association of Operational Investigation*, Bragança (Portugal)

- Brásio, A. S., Romanenko, A., and Fernandes, N. C. (2015c). Using sequential quadratic programming for system identification. *Applied Mathematics & Information Sciences*, 9(1):19–26. URL <http://www.naturalspublishing.com/Article.asp?ArtcID=7413>
- Brásio, A. S., Romanenko, A., and Fernandes, N. C. (2015a). Development of a numerically efficient biodiesel decanter simulator. In *Operational Research*, CIM Series in Mathematical Sciences 4. Springer International Publishing Switzerland 2015. URL <http://www.springer.com/us/book/9783319203270>

### Control Loop Performance Assessment

- Brásio, A. S. R., Romanenko, A., and Fernandes, N. C. P. (2014). Modeling, detection and quantification, and compensation of stiction in control loops: The state of the art. *Industrial & Engineering Chemistry Research*, 53(39):15020–15040. URL <http://dx.doi.org/10.1021/ie501342y>
- Brásio, A. S. R., Romanenko, A., and Fernandes, N. C. P. (2014). Stiction detection and quantification as an application of optimization. In Murgante, B., Misra, S., Rocha, A., Torre, C., Rocha, J., Falcão, M., Taniar, D., Apduhan, B., and Gervasi, O., editors, *Computational Science and Its Applications – ICCSA 2014*, volume 8580 of *Lecture Notes in Computer Science*, pages 169–179. Springer International Publishing. URL [http://dx.doi.org/10.1007/978-3-319-09129-7\\_13](http://dx.doi.org/10.1007/978-3-319-09129-7_13)
- Brásio, A. S., Romanenko, A., and Fernandes, N. C. (2015). Detection of stiction in level control loops. *IFAC-PapersOnLine*, 48(8):421 – 426. 9th IFAC Symposium on Advanced Control of Chemical Processes ADCHEM 2015 Whistler, Canada, June 7-10, 2015. URL <http://www.sciencedirect.com/science/article/pii/S240589631501085X>

- Romanenko, A., Brásio, A. S., Fernandes, N. C., Portugal, A. A., and Santos, L. O. (2015). Método e sistema de análise do desempenho de anéis de controlo industriais pela deteção e caracterização automáticas de oscilações múltiplas. URL [http://worldwide.espacenet.com/publicationDetails/biblio?DB=worldwide.espacenet.com&II=21&ND=3&adjacent=true&locale=en\\_EP&FT=D&date=20150318&CC=PT&NR=107168A&KC=A](http://worldwide.espacenet.com/publicationDetails/biblio?DB=worldwide.espacenet.com&II=21&ND=3&adjacent=true&locale=en_EP&FT=D&date=20150318&CC=PT&NR=107168A&KC=A). Patent Application PT 107168 (A)
- Brásio, A. S., Romanenko, A., and Fernandes, N. C. (2015b). Performance monitoring of industrial process control loops. In *Proceedings of the XVI Convention Of Electrical Engineering, CIE 2015, Villa Clara, (Cuba)*

### PID controller tuning

- Romanenko, A., Brásio, A. S., Fernandes, N. C., Portugal, A. A., and Santos, L. O. (2014). Monitorização e optimização do desempenho de controladores na presença de falhas nos elementos finais de controlo. URL [http://worldwide.espacenet.com/publicationDetails/biblio?DB=worldwide.espacenet.com&II=0&ND=3&adjacent=true&locale=en\\_EP&FT=D&date=20140630&CC=PT&NR=106716A&KC=A](http://worldwide.espacenet.com/publicationDetails/biblio?DB=worldwide.espacenet.com&II=0&ND=3&adjacent=true&locale=en_EP&FT=D&date=20140630&CC=PT&NR=106716A&KC=A). Patent Application PT 106716 (A)

## 6.2 Future work

Although the topics covered in this thesis appear to be mature areas, further refinements are desirable. Based on the reflection and inspiration from this thesis, several potential research directions may be provided.

### 6.2.1 System Identification

**On-line and automatic time delay estimation to improve reliability of CLPA tools:** Several methods are strictly dependent on the time delay of the process such as the Harris index. Usually, its estimation is not performed in the common practice and a default value is used which lowers the effectiveness of the methods

to detect and diagnose the performance problems of a control loop. In this context, the on-line and automatic time delay estimation would be of extreme importance to improve the reliability of control loop performance assessment tools. Although several univariable and multivariable methodologies are available in literature, an investigation over the existing methodologies to identify those which perform better in data containing noise, setpoint changes, external disturbances, and other phenomena would be very interesting. For several default scenarios, this study could identify the time delay estimation methods robustness and automatically separate them in groups. Then, several improvements could be suggested based on the weaknesses revealed by each of the groups.

### 6.2.2 Control Loop Performance Assessment

#### **Compensation of stiction considering the multivariable system performance:**

Although some emphasis of this thesis was placed on the refinement of stiction detection, quantification, and compensation methods, other improvements may be developed. For instance, stiction compensation could be viewed as a more general active fault-tolerant control problem that handles multivariable nonlinear systems. This way, instead of mitigating the effect of stiction in a single valve, a more efficient closed-loop performance may be achieved by taking advantage of eventual analytical and hardware redundancy. An example of such approach may be the use of online stiction diagnosis built into the MPC system and the adaptation of the move suppression and weight parameters associated to the manipulated variables.

#### **Increasing precision/accuracy of stiction detection and diagnosis results:**

Even though stiction detection and diagnosis methods based on model fitting have proved to be very efficient, they are difficult to implement in an automatic way. It is therefore desirable to develop an unified approach that combines several stiction diagnosis methods in order to increase the robustness, precision, and accuracy.

**Translating performance improvement to economic benefits:** When performance degradation is identified, corrective measures are applied in order to bring it back to the desired level. However, the common practice is to base the evaluation using purely technical terms without any reflection on the process eco-

nomics. This economic benefit analysis may show what impact a particular loop has and help prioritize loop maintenance and other optimization initiatives. Besides, the use of economic indicators may allow to provide more transparent information to various plant stakeholders and increase their confidence in CLPA tools.

**Holistic view of plant goals:** The interest in the performance measurement and optimization at each plant layer is high. However, these layers may have different goals and metrics that may be conflicting and mismatching, such as keeping the inventory low but asking to be ready for meeting a higher demand on the production. This problem could be solved by developing a platform that provides a holistic view of performance metrics across the whole plant. This would harmonize these metrics and focus the plant layers on meeting the company goals.

**Dissemination of advanced CLPA tools in industrial context:** For the last few years, commercial tools dedicated to the analysis of performance have emerged. However, these tools still lack the more recent advanced metrics that help detect, diagnose, and compensate control loop performance degradation. In addition, it may be beneficial to incorporate automatic functionalities that, in addition to the diagnosis, would implement corrective measures, minimizing the performance loss. Besides, a wider dissemination of CLPA tools should be done both in the industry and academia.

### 6.2.3 PID controller tuning

**Evaluate performance of the developed method in other valve faults:** In this thesis, a method to compensate faults present in control valves through the automatic tuning of the PID controller and a successful application to a valve affected by stiction was performed. This method may be further generalized to other valve faults in order to evaluate its performance in the fault compensation.

### 6.2.4 Soft sensor technology

**Automatic selection of sufficient informative data from plant historians:** Data quality is critical for the success of soft sensor development. Therefore, an approach to extract automatically information rich data from the plant historian

could improve the quality of the soft sensors and reduce the development time. Shardt (2014) developed a first approach composed by two phases: the data segmentation that identifies regions that may be described using the same model, and the data quality assessment in which each region information content is analyzed. This method may be extended for multivariable systems.



# Bibliography

- Åström, K. and Hägglund, T. (2006). *Advanced PID control*. ISA-The Instrumentation, Systems, and Automation Society.
- Åström, K., Panagopoulos, H., and Hägglund, T. (1998). Design of PI controllers based on non-convex optimization. *Automatica*, 34(5):585 – 601. URL <http://www.sciencedirect.com/science/article/pii/S0005109898000119>.
- Abonyi, J., Babuska, R., Szeifert, F., Nagy, L., and Verbruggen, H. (2000). Design and application of block-oriented fuzzy models — fuzzy hammerstein model. In Suzuki, Y., Ovaska, S., Furuhashi, T., Roy, R., and Dote, Y., editors, *Soft Computing in Industrial Applications*, pages 79–88. Springer London. URL [http://link.springer.com/chapter/10.1007%2F978-1-4471-0509-1\\_7](http://link.springer.com/chapter/10.1007%2F978-1-4471-0509-1_7).
- Akaike, H. (1969). Fitting autoregressive models for prediction. *Annals of the Institute of Statistical Mathematics*, 21(1):243–247. URL <http://dx.doi.org/10.1007/BF02532251>.
- Alemohammad, M. (2011). Detection and compensation for stiction in multi-loop control systems. Master’s thesis, University of Alberta, Alberta (Canada). URL [https://era.library.ualberta.ca/public/datastream/get/uuid:2029342a-20a8-458a-9cd6-a46f164c300a/DS1/Alemohammad\\_%20Mahdi\\_%20Spring2011.pdf](https://era.library.ualberta.ca/public/datastream/get/uuid:2029342a-20a8-458a-9cd6-a46f164c300a/DS1/Alemohammad_%20Mahdi_%20Spring2011.pdf).
- Alemohammad, M. and Huang, B. (2011). Frequency analysis and experimental validation for stiction phenomenon in multi-loop processes. *Journal of Process Control*, 21(4):437–447. URL <http://www.sciencedirect.com/science/article/pii/S0959152411000357>.
- Alemohammad, M. and Huang, B. (2012). Compensation of control valve stiction through controller tuning. *Journal of Process Control*, 22(9):1800–1819. URL <http://www.sciencedirect.com/science/article/pii/S095915241200193X>.

## Bibliography

---

- Alhoniemi, E., Hollmén, J., Simula, O., and Vesanto, J. (1999). Process monitoring and modeling using the self-organizing map. *Integrated Computer Aided Engineering*, 6:3–14. URL <http://www.cis.hut.fi/projects/ide/publications/papers/jicae.ps>.
- Altpeter, F. (1993). *Friction modeling, identification and compensation*. PhD thesis, École Polytechnique Fédérale de Lausanne, Lausanne (Switzerland).
- Altpeter, F., Grunenberg, M., Myszkowski, P., and Longchamp, R. (2000). Auto-tuning of feedforward friction compensation based on the gradient method. *Proceedings of the American Control Conference*, 4:2600–2604. URL <http://dx.doi.org/10.1109/ACC.2000.878677>.
- Amin, J., Friedland, B., and Harnoy, A. (1997). Implementation of a friction estimation and compensation technique. *Proceedings of the IEEE Conference on Control Systems*, 17(4):71–76. URL <http://dx.doi.org/10.1109/37.608554>.
- Aransiola, E., Ojumu, T., Oyekola, O., Madzimbamuto, T., and Ikhu-Omoregbe, D. (2014). A review of current technology for biodiesel production: State of the art. *Biomass and Bioenergy*, 61:276 – 297.
- Araujo, A. P., Munaro, C. J., and Rosado Filho, M. (2012). Quantification of valve stiction and dead band in control loops based on the harmonic balance method. *Industrial & Engineering Chemistry Research*, 51(43):14121–14134.
- Armstrong Hérouvry, B., Dupont, P., and Canudas de Wit, C. (1994). A survey of models, analysis tools and compensation methods for the control of machines with friction. *Automatica*, 30(7):1083–1138. URL <http://www.sciencedirect.com/science/article/pii/0005109894902097>.
- Arumugam, Srinivasan; Panda, R. C. (2014). Identification of stiction nonlinearity for pneumatic control valve using anfis method. *International Journal of Engineering & Technology*, 6(2):570–578.
- Arumugam, S. and Panda, R. (2011). Control valve stiction identification, modelling, quantification and control -a review. *Sensors & Transducers Journal*, 132(9):14 – 24. International Frequency Sensor Association. HighBeam Research.
- Aspen Tech (2012). Aspen Watch Performance Monitor. URL <http://www.aspentech.com/products/aspen-watch.aspx>. Accessed on September 2012.



- Åström, K. (1970). *Introduction to stochastic control theory*. Mathematics in science and engineering. Academic Press.
- Åström, K. and Hägglund, T. (2004). Revisiting the Ziegler-Nichols step response method for PID control. *Journal of Process Control*, 14(6):635 – 650. MIGO.
- Atkeson, C., Moore, A., and Schaal, S. (1997). Locally weighted learning. *Artificial Intelligence Review*, 11(1-5):11–73.
- Babji, S., Nallasivam, U., and Rengaswamy, R. (2012). Root cause analysis of linear closed-loop oscillatory chemical process systems. *Ind. Eng. Chem. Res.*, 51(42):13712–13731.
- Bacci di Capaci, R. and Scali, C. (2014). Stiction quantification: A robust methodology for valve monitoring and maintenance scheduling. *Industrial & Engineering Chemistry Research*, 53(18):7507–7516.
- Baena-García, M., del Campo-Ávila, J., Fidalgo, R., Bifet, A., Gavaldá, R., and Morales-Bueno, R. (2006). Early drift detection method. In *In Fourth International Workshop on Knowledge Discovery from Data Streams*.
- Bakke, M. (2009). Subspace identification using closed-loop data. Master's thesis, Norwegian University of Science and Technology, Norway.
- Bambase, M., Nakamura, N., Tanaka, J., and Matsumura, M. (2007). Kinetics of hydroxide-catalyzed methanolysis of crude sunflower oil for the production of fuel-grade methyl esters. *Journal of Chemical Technology & Biotechnology*, 82(3):273–280. URL <http://onlinelibrary.wiley.com/doi/10.1002/jctb.1666/abstract>.
- Bauer, M., Cox, J. W., Caveness, M. H., Downs, J. J., and Thornhill, N. F. (2007). Finding the direction of disturbance propagation in a chemical process using transfer entropy. *Control Systems Technology, IEEE Transactions on*, 15(1):12 –21.
- Bauer, M., Thornhill, N., and Meaburn, A. (2004). Specifying the directionality of fault propagation paths using transfer entropy. In *7th International Symposium on Dynamics and Control of Process System, DYCOPS 7 Conference, Cambridge, Massachusetts, USA*.
- Beck, J. and Arnold, K. (1977). *Parameter estimation in engineering and science*. Wiley series in probability and mathematical statistics. Probability and mathematical statistics. Wiley.

## Bibliography

---

- Belanger, P. W. and Luyben, W. L. (1996). A new test for evaluation of the regulatory performance of controlled processes. *Industrial & Engineering Chemistry Research*, 35(10):3447–3457.
- Bell, B. M. (2012). CppAD: a package for C++ algorithmic differentiation. *Computational Infrastructure for Operations Research*, COIN-OR. URL <http://www.coin-or.org/CppAD>.
- Belli, P., Bonavita, N., and Rea, R. (2006). An integrated environment for industrial control performance assessment, diagnosis and improvement. In *Proceedings of the 2006 International Congress on Methodologies for Emerging Technologies in Automation ANIPLA, (Rome) Italy*, Rome.
- Bersimis, S., Psarakis, S., and Panaretos, J. (2007). Multivariate statistical process control charts: an overview. *Quality and Reliability Engineering International*, 23(5):517–543.
- Bezergianni, S. and Georgakis, C. (2000). Controller performance assessment based on minimum and open-loop output variance. *Control Engineering Practice*, 8(7):791 – 797.
- Bezergianni, S. and Georgakis, C. (2003). Evaluation of controller performance—use of models derived by subspace identification. *International Journal of Adaptive Control and Signal Processing*, 17(7-9):527–552.
- Biegler, L. (2007). *Real-time PDE-constrained optimization*. Computational science and engineering. Society for Industrial and Applied Mathematics.
- Bishop, C. (1995). *Neural Networks for Pattern Recognition*. Clarendon Press.
- Björklund, S. (2003). *A survey and comparison of time-delay estimation methods in linear systems*. PhD thesis, Linköping University, Linköping, Sweden.
- Bjorklund, S. and Ljung, L. (2003). A review of time-delay estimation techniques. In *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, volume 3, pages 2502 – 2507 Vol.3.
- Bliman, P. and Sorine, M. (1995). Easy-to-use realistic dry friction models for automatic control. *Proceedings of the European Control Conference*, pages 3788–3794. URL <https://who.rocq.inria.fr/Pierre-Alexandre.Bliman/D4.pdf.zip>.
- Bobál, V. (2005). *Digital Self-tuning Controllers: Algorithms, Implementation and Applications*. Advanced Textbooks in Control and Signal Processing. Springer.

- Boizot, N., Busvelle, E., and Gauthier, J.-P. (2010). An adaptive high-gain observer for nonlinear systems. *Automatica*, 46(9):1483 – 1488.
- Bolstad, W. (2004). *Introduction to Bayesian Statistics*. Wiley.
- Bona, B. and Indri, M. (2005). Friction compensation in robotics: an overview. *Proceedings of the European Control Conference*, pages 4360–4367. URL <http://dx.doi.org/10.1109/CDC.2005.1582848>.
- Bonavita, N., Bovero, J. C., Martini, R., and Lorenzo, d. L. (2006). Control performance monitoring: a data-driven approach. In *Proceedings of the ANIPLA Workshop, Milan (Italy)*, Milan.
- Bottou, E. and Vapnik, V. (1992). Local learning algorithms. *Neural Computation*, 4:888–900.
- Box, G., Jenkins, G., and Reinsel, G. (1970). *Time series analysis: forecasting and control*. Forecasting and Control Series. Prentice Hall.
- Box, G. and Tiao, G. (1973). *Bayesian inference in statistical analysis*. Addison-Wesley series in behavioral science: quantitative methods. Addison-Wesley Pub. Co.
- Brásio, A. S., Romanenko, A., and Fernandes, N. C. (2012). System identification as an application of optimization. In Simos, T. E., Psihoyios, G., Tsitouras, C., and Anastassi, Z., editors, *Proceedings of the AIP Conference*, volume 1479, pages 822–825, Kos (Greece). AIP. URL <http://dx.doi.org/10.1063/1.4756264>.
- Brásio, A. S., Romanenko, A., and Fernandes, N. C. (2013). Aproximação de cálculos iterativos por redes neuronais em sistemas de equações diferenciais ordinárias. In *Proceedings of the XVI Congress of the Portuguese Association of Operational Investigation*, Bragança (Portugal).
- Brásio, A. S., Romanenko, A., and Fernandes, N. C. (2015a). Development of a numerically efficient biodiesel decanter simulator. In *Operational Research, CIM Series in Mathematical Sciences 4*. Springer International Publishing Switzerland 2015. URL <http://www.springer.com/us/book/9783319203270>.
- Brásio, A. S., Romanenko, A., and Fernandes, N. C. (2015b). Performance monitoring of industrial process control loops. In *Proceedings of the XVI Convention Of Electrical Engineering, CIE 2015, Villa Clara, (Cuba)*.

## Bibliography

---

- Brásio, A. S., Romanenko, A., and Fernandes, N. C. (2015c). Using sequential quadratic programming for system identification. *Applied Mathematics & Information Sciences*, 9(1):19–26. URL <http://www.naturalspublishing.com/Article.asp?ArtcID=7413>.
- Brásio, A. S., Romanenko, A., Fernandes, N. C., and Santos, L. O. (2015). First principle modeling and predictive control of a continuous biodiesel plant. *Journal of Process Control*, pages 1–33. Submitted paper.
- Brásio, A. S., Romanenko, A., Leal, J., Santos, L. O., and Fernandes, N. C. (2013). Non-linear model predictive control of biodiesel production via transesterification of used vegetable oils. *Journal of Process Control*, 23(10):1471 – 1479. URL <http://dx.doi.org/10.1016/j.jprocont.2013.09.023>.
- Brásio, A. S., Romanenko, A., Santos, L. O., and Fernandes, N. C. (2011). Modeling the effect of mixing in biodiesel production. *Bioresource Technology*, 102(11):6508 – 6514. URL <http://www.sciencedirect.com/science/article/pii/S0960852411004470>.
- Brásio, A. S. R., Romanenko, A., and Fernandes, N. C. P. (2014). Modeling, detection and quantification, and compensation of stiction in control loops: The state of the art. *Industrial & Engineering Chemistry Research*, 53(39):15020–15040. URL <http://dx.doi.org/10.1021/ie501342y>.
- Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24(2):123–140.
- Brito, A. G. (2011). Computation of multiple limit cycles in nonlinear control systems – A describing function approach. *Journal of Aerospace of Technology and Management*, 3(1):21–28. URL <http://dx.doi.org/10.5028/jatm.2011.03017010>.
- Brockwell, P. and Davis, R. (2002). *Introduction to Time Series and Forecasting*. Lecture Notes in Statistics. Springer.
- Brásio, A. S., Romanenko, A., and Fernandes, N. C. (2015). Detection of stiction in level control loops. *IFAC-PapersOnLine*, 48(8):421 – 426. 9th IFAC Symposium on Advanced Control of Chemical Processes ADCHEM 2015 Whistler, Canada, June 7-10, 2015.
- Brásio, A. S. R., Romanenko, A., and Fernandes, N. C. P. (2014). Stiction detection and quantification as an application of optimization. In Murgante, B., Misra, S., Rocha, A., Torre, C., Rocha, J., Falcão, M., Tanir, D., Apduhan, B., and Gervasi, O., editors,

- Computational Science and Its Applications – ICCSA 2014*, volume 8580 of *Lecture Notes in Computer Science*, pages 169–179. Springer International Publishing. URL [http://dx.doi.org/10.1007/978-3-319-09129-7\\_13](http://dx.doi.org/10.1007/978-3-319-09129-7_13).
- Buckbee, G. (2008). The 6 most common PID configuration errors: How to find and fix them. URL <http://www.expertune.com/articles/WPPIDConfigErrors.pdf>. Consulted: May 2015.
- Campestrini, L., Filho, L., and Bazanella, A. (2009). Tuning of multivariable decentralized controllers through the ultimate-point method. *Control Systems Technology, IEEE Transactions on*, 17(6):1270–1281.
- Canudas de Wit, C. (1993). Robust control for servo-mechanisms under inexact friction compensation. *Automatica*, 29(3):757–761. URL <http://www.sciencedirect.com/science/article/pii/000510989390070A>.
- Canudas de Wit, C. and Lischinsky, P. (1997). Adaptive friction compensation with partially known dynamic friction model. *International Journal of Adaptive Control and Signal Processing*, 11(1):65–80. URL [http://dx.doi.org/10.1002/\(SICI\)1099-1115\(199702\)11:1<65::AID-ACS395>3.0.CO;2-3](http://dx.doi.org/10.1002/(SICI)1099-1115(199702)11:1<65::AID-ACS395>3.0.CO;2-3).
- Canudas de Wit, C., Olsson, H., Åström, K., and Lischinsky, P. (1995). A new model for control of systems with friction. *IEEE Transactions on Automatic Control*, 40(3):419–425. URL <http://dx.doi.org/10.1109/9.376053>.
- Canudas de Wit, C., Åström, K., and Braun, K. (1987). Adaptive friction compensation in DC-motor drives. *IEEE Journal of Robotics and Automation*, 3(6):681–685. URL <http://dx.doi.org/10.1109/JRA.1987.1087142>.
- Chandrashekar, G. and Sahin, F. (2014). A survey on feature selection methods. *Computers & Electrical Engineering*, 40(1):16–28. URL <http://www.sciencedirect.com/science/article/pii/S0045790613003066>.
- Chaturvedi, D. (2008). *Soft Computing: Techniques and Its Applications in Electrical Engineering*. Studies in Computational Intelligence. Springer.
- Chen, D. and Seborg, D. E. (2003). Design of decentralized pi control systems based on Nyquist stability analysis. *Journal of Process Control*, 13(1):27–39.

## Bibliography

---

- Chen, S.-L. (2009). *Modeling of precision motion control systems: a relay feedback approach*. PhD thesis, National University of Singapore, Singapore (Republic of Singapore). URL <http://scholarbank.nus.edu.sg/handle/10635/16548>.
- Chen, S.-L., Tan, K. K., and Huang, S. (2008). Two-layer binary tree data-driven model for valve stiction. *Ind. Eng. Chem. Res.*, 47(8):2842–2848.
- Chiang, R. C. and Yu, C. C. (1993). Monitoring procedure for intelligent control: on-line identification of maximum closed-loop log modulus. *Industrial & Engineering Chemistry Research*, 32(1):90–99.
- Chitralkha, S. B., Shah, S. L., and Prakash, J. (2010). Detection and quantification of valve stiction by the method of unknown input estimation. *Journal of Process Control*, 20(2):206 – 216.
- Choi, S. W., Martin, E. B., Morris, A. J., and Lee, I.-B. (2006). Adaptive multivariate statistical process control for monitoring time-varying processes. *Industrial & Engineering Chemistry Research*, 45(9):3108–3118.
- Chokshi, N. K. (2012). Soft sensors development for model based control of groundwater treatment plants. Master's thesis, Delft University of Technology, Netherlands.
- Choudhury, A. A. S., Shah, S., Thornhill, N. F., and Shook, D. (2007). Detection and quantification of stiction. US Patent 2007/0288103 A1.
- Choudhury, A. A. S., Shah, S. L., and Thornhill, N. F. (2012). Detection and quantification of stiction. US Patent 8145328.
- Choudhury, A. A. S., Shah, S. L., Thornhill, N. F., and Shook, D. S. (2006a). Automatic detection and quantification of stiction in control valves. *Control Engineering Practice*, 14(12):1395–1412. URL <http://www.sciencedirect.com/science/article/pii/S0967066105002388>.
- Choudhury, A. A. S., Thornhill, N. F., and Shah, S. L. (2005). Modelling valve stiction. *Control Eng. Pract.*, 13(5):641–658.
- Choudhury, M. (2011). Plantwide oscillations diagnosis – current state and future directions. *Asia-Pacific Journal of Chemical Engineering*, 6(3):484–496.
- Choudhury, M., Jain, M., and Shah, S. (2006b). Detection and quantification of valve stiction. In *American Control Conference, 2006*, page 10 pp.

- Choudhury, M. S., Jain, M., and Shah, S. L. (2008a). Stiction – definition, modelling, detection and quantification. *Journal of Process Control*, 18:232 – 243. Festschrift honouring Professor Dale Seborg.
- Choudhury, M. S., Shah, S., and Thornhill, N. (2004a). Diagnosis of poor control-loop performance using higher-order statistics. *Automatica*, 40(10):1719 – 1728.
- Choudhury, M. S., Thornhill, N. F., and Shah, S. L. (2004b). A data-driven model for valve stiction.
- Choudhury, S., Jain, M., and Shah, S. L. (2008b). Stiction – Definition, modelling, detection and quantification. *J. Process Contr.*, 18:232–243.
- Ciengis (2015). Plantstreamer Portal. URL <http://www.ciengis.com/products/plantstreamer-portal/>. Accessed in September 2015.
- Cinar, A. and Undey, C. (1999). Statistical process and controller performance monitoring, a tutorial on current methods and future directions. In *American Control Conference, 1999. Proceedings of the 1999*, volume 4, pages 2625 –2639 vol.4, San Diego - California.
- Cohen, G. H. and Coon, G. A. (1953). Theoretical consideration of retarded control. *Transactions of the ASME*, 75:827 – 834.
- Control Arts (2012). Controller performance assessment for industrial control loops. URL <http://www.controlartsinc.com/ControlMonitor/ControlMonitor.html>. Accessed on September 2012.
- ControlSoft (2012). INTUNE. URL <http://www.controlsoftinc.com/intune5plus.shtml>. Accessed on August 2012.
- Dahl, P. (1968). A solid friction model. Technical report, Aerospace Corporation, El Segundo (Canada). No. TOR-0158(3107-18)-1. URL [www.dtic.mil/dtic/tr/fulltext/u2/a041920.pdf](http://www.dtic.mil/dtic/tr/fulltext/u2/a041920.pdf).
- Dahlin, E. B. (1968). Designing and tuning digital controllers. *Instruments and Control Systems*, 41(6):77 – 83.
- Dai, C. and Yang, S. H. (2004). Controller performance assessment with a lqg benchmark obtained by using the subspace method. In Sahinkaya, M. N. and Edge, K. A., editors, *Proceedings of the Control 2004 Conference*. University of Bath, U.K.

## Bibliography

---

- Danesh Pour, N., Huang, B., and Shah, S. (2009). Consistency of noise covariance estimation in joint input–output closed-loop subspace identification with application in lqg benchmarking. *Journal of Process Control*, 19(10):1649 – 1657.
- Davies, L. and Gather, U. (1993). The identification of multiple outliers. *Journal of the American Statistical Association*, 88(423):782–792.
- Dayal, B. S. and MacGregor, J. F. (1996). Identification of finite impulse response models: Methods and robustness issues. *Industrial & Engineering Chemistry Research*, 35(11):4078–4090.
- de Souza L. Cuadros, M., Munaro, C., and Munareto, S. (2010). An improved algorithm for automatic quantification of valve stiction in flow control loops. In *Industrial Technology (ICIT), 2010 IEEE International Conference on*, pages 173 –178.
- de Souza L. Cuadros, M. A., Munaro, C. J., and Munareto, S. (2012a). Improved stiction compensation in pneumatic control valves. *Computers & Chemical Engineering*, 38(0):106 – 114.
- de Souza L. Cuadros, M. A., Munaro, C. J., and Munareto, S. (2012b). Novel model-free approach for stiction compensation in control valves. *Industrial & Engineering Chemistry Research*, 51(25):8465–8476.
- Deflorian, M. and Zaglauer, S. (2011). Design of experiments for nonlinear dynamic system identification. In *Proceedings of the 18th IFAC World Congress*, volume 18. URL <http://www.ifac-papersonline.net/Detailed/51801.html>.
- Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 39(1).
- Dennis, J. and Schnabel, R. (1983). *Numerical methods for unconstrained optimization and nonlinear equations*. Prentice-Hall series in computational mathematics. Prentice-Hall.
- 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.



- Desborough, L. and Miller, R. (2002). Increasing customer value of industrial control performance monitoring – honeywell’s experience.
- 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.
- DiBella, A., Fortuna, L., Graziani, S., Napoli, G., and Xibilia, M. (2007). A comparative analysis of the influence of methods for outliers detection on the performance of data driven models. In *Proceedings of the IEEE Conference on Instrumentation and Measurement Technology*, pages 1–5.
- Ding, F. and Chen, T. (2005). Modeling and identification of multirate systems. *Acta Automatica Sinica*, 31(1):105 – 122.
- Dittmar, R., Gill, S., Singh, H., and Darby, M. (2012). Robust optimization-based multi-loop PID controller tuning: A new tool and its industrial application. *Control Engineering Practice*, 20(4):355 – 370. Special Section: IFAC Symposium on Advanced Control of Chemical Processes - ADCHEM 2009.
- Dixon, W., Dixon, W., and Brown, B. (1988). *BMDP Statistical Software Manual: To Accompany the 1988 Software Release*. Number v. 1 in *BMDP Statistical Software Manual: To Accompany the 1988 Software Release*. University of California Press.
- Domingos, P. and Hulten, G. (2000). Mining high-speed data streams. In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '00*, pages 71–80, New York, NY, USA. ACM.
- Doraiswami, R. and Cheded, L. (2014). Robust model-based soft sensor: design and application. In *The International Federation of Automatic Control*.
- Dow (2015). Glycerol purification from biodiesel processing. URL [http://www.dowwaterandprocess.com/en/industries-and-applications/chemical\\_and\\_petrochemical/biodiesel/glycerol\\_purification](http://www.dowwaterandprocess.com/en/industries-and-applications/chemical_and_petrochemical/biodiesel/glycerol_purification), accessed on July 2015.
- Dries, A. and Rückert, U. (2009). Adaptive concept drift detection. *Statistical Analysis and Data Mining*, 2(5-6):311–327. URL <http://dx.doi.org/10.1002/sam.10054>.

## Bibliography

---

- Du, X., Liu, L., Xi, X., Yang, L., Yang, Y., Liu, Z., Zhang, X., Yu, C., and Du, J. (2011). Back pressure prediction of the direct air cooled power generating unit using the artificial neural network model. *Applied Thermal Engineering*, 31(14-15):3009 – 3014. URL <http://www.sciencedirect.com/science/article/pii/S1359431111002948>.
- Dumont, G. A., Kammer, L., Allison, B. J., Ettaleb, L., and Roche, A. A. (2002). Control system performance monitoring: New developments and practical issues. In *Proceedings of the 15th IFAC World Congress, Barcelona (Spain)*, volume 15(1), pages 1616 – 1616, Barcelona, Spain.
- Dunia, R. and Qin, S. J. (1998). Joint diagnosis of process and sensor faults using principal component analysis. *Control Engineering Practice*, 6(4):457 – 469.
- Dupont, P., Armstrong, B., and Hayward, V. (2000). Elasto-plastic friction model: contact compliance and stiction. *Proceedings of the American Control Conference*, 2:1072–1077. URL <http://dx.doi.org/10.1109/ACC.2000.876665>.
- Dupont, P., Hayward, V., Armstrong, B., and Altpeter, F. (2002). Single state elastoplastic friction models. *IEEE Transactions on Automatic Control*, 47(5):787–792. URL <http://dx.doi.org/10.1109/TAC.2002.1000274>.
- Eaton, J. W. (2002). *Gnu octave manual*. Network Theory Limited.
- Emerson (2012). Control Performance Tools. URL <http://www2.emersonprocess.com/en-US/brands/processautomation/consultingservices/ProcessImprovementOptimization/ControlPerformanceServices/Pages/ControlPerformanceTools.aspx>. Accessed on September 2012.
- Enders, C. (2010). *Applied Missing Data Analysis*. Methodology in the social sciences. Guilford Press.
- Enzner, G. (2010). Bayesian inference model for applications of time-varying acoustic system identification. In *European Signal Processing Conference (EUSIPCO)*, Aalborg (Denmark). URL <http://www.eurasip.org/Proceedings/Eusipco/Eusipco2010/Contents/papers/1569291461.pdf>.
- Eren-Oruklu, M., Cinar, A., Rollins, D. K., and Quinn, L. (2012). Adaptive system identification for estimating future glucose concentrations and hypoglycemia alarms. *Automatica*, 48(8):1892–1897.

- Eriksson, L. (2008). *PID controller design and tuning in networked control systems*. PhD thesis, Helsinki University of Technology Control Engineering.
- Eriksson, L. and Koivo, H. (2005). Tuning of discrete-time PID controllers in sensor network based control systems. In *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation, CIRA '05*, pages 359–364, Espoo (Finland).
- 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.
- Escobar, H. J. (2012). *Advanced Monitoring and Soft Sensor Development with Application to Industrial Processes*. PhD thesis, Auburn University.
- Eskinat, E., Johnson, S. H., and Luyben, W. L. (1991). Use of Hammerstein models in identification of nonlinear systems. *AIChE J.*, 37(2):255–268.
- Ettaleb, L. (1999). *Control loop performance assessment and oscillation detection*. PhD thesis, University of British Columbia, Vancouver, Canada.
- ExpertTune (2012). Comparison of PID control algorithms. URL <http://www.experttune.com/artCE87.aspx>. Accessed on April 2012.
- Fang, L., Wang, J., Tan, X., and Shang, Q. (2015). Analysis and compensation of control valve stiction-induced limit cycles. In *54th IEEE Conference on Decision and Control*. Paper ID: 55. Submitted for publication.
- Farenzena, M. (2008). *Novel methodologies for assessment and diagnostics in control loop management*. PhD thesis, Universidade Federal do Rio Grande do Sul, Porto Alegre (Brasil).
- Farenzena, M. and Trierweiler, J. (2012). Valve stiction estimation using global optimisation. *Control Engineering Practice*, 20(4):379–385. Special Section: IFAC Symposium on Advanced Control of Chemical Processes – ADCHEM 2009.
- Farenzena, M. and Trierweiler, J. O. (2010). Modified PI controller for stiction compensation. *Proceedings of the International Symposium on Dynamics and Control of Process Systems*, pages 791–796. URL [http://www.nt.ntnu.no/users/skoge/prost/proceedings/dycops-2010/Papers\\_DYCOPS2010/WeET1-04.pdf](http://www.nt.ntnu.no/users/skoge/prost/proceedings/dycops-2010/Papers_DYCOPS2010/WeET1-04.pdf).

## Bibliography

---

- Farenzena, M. and Trierweiler, J. Q. (2012). Valve backlash and stiction detection in integrating processes. *Proceedings of the International Federation of Automatic Control*, pages 320–324. URL <http://dx.doi.org/10.3182/20120710-4-SG-2026.00152>.
- Fauvel, M., Chanussot, J., and Benediktsson, J. (2006). Kernel principal component analysis for feature reduction in hyperspectral images analysis. In *Signal Processing Symposium, 2006. NORSIG 2006. Proceedings of the 7th Nordic*, pages 238–241.
- Feemster, M., Vedagarbha, P., Dawson, D., and Haste, D. (1998). Adaptive control techniques for friction compensation. *Proceedings of the American Control Conference*, 3:1488–1492. URL <http://dx.doi.org/10.1109/ACC.1998.707076>.
- Ferretti, G., Magnani, G., and Rocco, P. (2004). Single and multistate integral friction models. *IEEE Transactions on Automatic Control*, 49(12):2292–2297. URL <http://dx.doi.org/10.1109/TAC.2004.839234>.
- Flowserve (2012). Monitoring and controls. URL <http://www.flowserve.com/Products/Monitoring-and-Controls>. Accessed on September 2012.
- Forsman, K. and Stattin, A. (1999). A new criterion for detecting oscillations in control loops. In *Proceedings of the European Control Conference*, pages Karlsruhe, Germany.
- Forsell, U. (1999). *Closed-loop Identification : Methods, Theory, and Applications*. PhD thesis, Linköping University, Automatic Control, The Institute of Technology.
- Forsell, U. and Ljung, L. (1997). Issues in closed-loop identification. Technical report.
- Forsell, U. and Ljung, L. (1999). Closed-loop identification revisited. *Automatica*, 35(7):1215 – 1241.
- Fortescue, T., Kershenbaum, L., and Ydstie, B. (1981). Implementation of self-tuning regulators with variable forgetting factors. *Automatica*, 17(6):831 – 835.
- Fortuna, L., Giannone, P., Graziani, S., and Xibilia, M. (2007a). Virtual instruments based on stacked neural networks to improve product quality monitoring in a refinery. *Instrumentation and Measurement, IEEE Transactions on*, 56(1):95–101.
- Fortuna, L., Graziani, S., Rizzo, A., and Xibilia, M. (2007b). *Soft Sensors for Monitoring and Control of Industrial Processes*. Advances in Industrial Control. Springer.
- Fortuna, L., Graziani, S., and Xibilia, M. (2005a). Soft sensors for product quality monitoring in debutanizer distillation columns. *Control Engineering Practice*, 13(4):499 – 508.

- Fortuna, L., Graziani, S., and Xibilia, M. (2005b). Virtual instruments in refineries. *Instrumentation Measurement Magazine, IEEE*, 8(4):26–34.
- Fortuna, L., Graziani, S., and Xibilia, M. (2009). Comparison of soft-sensor design methods for industrial plants using small data sets. *Instrumentation and Measurement, IEEE Transactions on*, 58(8):2444–2451.
- Frau, A., Baratti, R., and Alvarez, J. (2009). Composition estimation of a six-component distillation column with temperature measurements. In *7th IFAC International Symposium on Advanced Control of Chemical Processes*.
- Frau, A., Baratti, R., and Alvarez, J. (2010). Measurement structure design for multi-component distillation column with specific estimation objective. In *9th International Symposium on Dynamics and Control of Process Systems*.
- Fredenslund, A., Jones, R. L., and Prausnitz, J. M. (1975). Group-contribution estimation of activity coefficients in nonideal liquid mixtures. *AIChE Journal*, 21(6):1086–1099. URL <http://onlinelibrary.wiley.com/doi/10.1002/aic.690210607/abstract;jsessionid=C19AE6A49C6BAB411A7C7B77DD43E0C1.f04t03>.
- Friedland, B. and Park, Y.-J. (1992). On adaptive friction compensation. *IEEE Transactions on Automatic Control*, 37(10):1609–1612. URL <http://dx.doi.org/10.1109/9.256395>.
- Fujiwara, K., Kano, M., Hasebe, S., and Takinami, A. (2009). Soft-sensor development using correlation-based just-in-time modeling. *AIChE Journal*, 55(7):1754–1765.
- Gama, J., Medas, P., Castillo, G., and Rodrigues, P. (2004). Learning with drift detection. In Bazzan, A. and Labidi, S., editors, *Advances in Artificial Intelligence – SBIA 2004*, volume 3171 of *Lecture Notes in Computer Science*, pages 286–295. Springer Berlin Heidelberg.
- Gama, J. a., Žliobaitė, I., Bifet, A., Pechenizkiy, M., and Bouchachia, A. (2014). A survey on concept drift adaptation. *ACM Comput. Surv.*, 46(4):44:1–44:37.
- Garcia, C. (2008). Comparison of friction models applied to a control valve. *Control Engineering Practice*, 16(10):1231–1243. URL <http://www.sciencedirect.com/science/article/pii/S0967066108000221>.
- Ge, Z., Chen, T., and Song, Z. (2011). Quality prediction for polypropylene production process based on {CLGPR} model. *Control Engineering Practice*, 19(5):423 – 432.

## Bibliography

---

- Geffen, V. (2009). A study of friction models and friction compensation. Master's thesis, Eindhoven University of Technology, Eindhoven (Netherlands). URL <http://www.mate.tue.nl/mate/pdfs/11194.pdf>.
- Gerry, J. and Ruel, M. (2001). How to measure and combat valve stiction online? *Proceedings of the ISA International Fall Conference*. URL <http://www.expertune.com/articles/isa2001/StictionMR.htm>.
- Gerry, J. P. (2002). Prioritizing and optimizing problem loops using a loop monitoring system. In *International Society of Automation 2002, Chicago*. expertune. URL [https://www.idc-online.com/technical\\_references/pdfs/instrumentation/Prioritizing%20and%20Optimizing%20Problem%20Loops%20Using%20a%20Loop%20monitoring%20system.pdf](https://www.idc-online.com/technical_references/pdfs/instrumentation/Prioritizing%20and%20Optimizing%20Problem%20Loops%20Using%20a%20Loop%20monitoring%20system.pdf).
- Ghomrawi, H., Mandl, L., Rutledge, J., Alexiades, M., and Mazumdar, M. (2011). Is there a role for expectation maximization imputation in addressing missing data in research using womac questionnaire? comparison to the standard mean approach and a tutorial. *BMC Musculoskeletal Disorders*, 12(1):109.
- GmbH, L. (2012). Plant loop auditing in practice. URL [www.leikon.de/downloads/atp\\_loop\\_auditing\\_akesson.pdf](http://www.leikon.de/downloads/atp_loop_auditing_akesson.pdf). Accessed on September 2012.
- Goldfeld, S. and Quandt, R. (1972). *Nonlinear methods in econometrics*. Contributions to economic analysis. North-Holland Pub. Co.
- Gomnam, E. and Jazayeri-rad, H. (2013). Development of an adaptive soft sensor based on fcmilssvr. *International Journal of Scientific & Technology Research*, 2(2).
- Gonzalez, G. (1999). Soft sensors for processing plants. In *Intelligent Processing and Manufacturing of Materials, 1999. IPMM '99. Proceedings of the Second International Conference on*, volume 1, pages 59–69 vol.1.
- Good, P. (2006). *Resampling Methods: A Practical Guide to Data Analysis*. Birkhäuser Boston.
- Goodwin, G., Graebe, S., and Salgado, M. (2001). *Control System Design*. Prentice Hall.
- Grazia ni, S., Napoli, G., and Xibilia, M. (2008). Soft sensor design for a sulfur recovery unit using a clustering based approach. In *Instrumentation and Measurement Technology Conference Proceedings, 2008. IMTC 2008. IEEE*, pages 1162–1167.

- Graziani, S., Pagano, F., and Xibilia, M. (2010). Soft sensor for a propylene splitter with seasonal variations. In *Instrumentation and Measurement Technology Conference (I2MTC), 2010 IEEE*, pages 273–278.
- Grimble, M. (2002). Controller performance benchmarking and tuning using generalised minimum variance control. *Automatica*, 38(12):2111 – 2119.
- Grimble, M. J. (2003). Robustness of full order and restricted structure optimal control systems. In *Proceedings of the 42nd IEEE Conference on Decision and Control*, pages 227 – 232, Maui, Hawaii, USA.
- Grimble, M. J. (2004). Gmv control of nonlinear multivariable systems. *IEE Control 2004 (Bath)*.
- Grimble, M. J. and Majecki, P. (2004). Weighting Selection for Controller Benchmarking and Tuning. Technical report, Industrial Control Centre - University of Strathclyde - Glasgow, Glasgow.
- Grubbs, F. E. (1969). Procedures for detecting outlying observations in samples. *Technometrics*, 11(1):1–21.
- Guidorzi, R. (2003). *Multivariable System Identification. From Observations to Models*. Bononia University Press.
- Guillet, J., Mourllion, B., Birouche, A., and Basset, M. (2011). Extracting second-order structures from single-input state-space models: application to model order reduction. *International Journal of Applied Mathematics and Computer Science*, 21(3):509–519.
- Gupta, M., Gao, J., Aggarwal, C., and Han, J. (2014). Outlier detection for temporal data: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 25(1):1–20.
- Gustafsson, F. (2000). *Adaptive filtering and change detection*. Wiley.
- Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182.
- György, A. and Kocsis, L. (2011). Efficient multi-start strategies for local search algorithms. *Journal of Artificial Intelligence Research*.
- Hagan, M., Demuth, H., and Beale, M. (1996). *Neural Network Design*. Electrical Engineering Series. Brooks/Cole.

## Bibliography

---

- Hägglund, T. (1997). Stiction compensation in control valves. *Proceedings of the European Control Conference*.
- Haggund, T. (1999). Automatic detection of sluggish control loops. *Control Engineering Practice*, 7(12):1505 – 1511.
- Haggund, T. (2005). Industrial implementation of on-line performance monitoring tools. *Control Engineering Practice*, 13(11):1383 – 1390.
- Hägglund, T. (2011). A shape-analysis approach for diagnosis of stiction in control valves. *Control Engineering Practice*, 19(8):782–789. URL <http://www.sciencedirect.com/science/article/pii/S0967066111000438>.
- Haibo, W. and Maying, Y. (2010). Performance assessment of cascade control system based on generalized minimum variance benchmarking. In *Control Conference (CCC), 2010 29th Chinese*, pages 3876–3881.
- Haimi, H., Mulas, M., Corona, F., and Vahala, R. (2013). Data-derived soft-sensors for biological wastewater treatment plants: An overview. *Environmental Modelling & Software*, 47(0):88 – 107.
- Halimi, B. and Kune, S. (2010). Analysis of nonlinearities compensation for control valves. *Proceedings of the International Congress on Advances in Nuclear Power Plants*. URL <https://getinfo.de/app/Analysis-of-Nonlinearities-Compensation-for-Control/id/BLCP%3ACN076690904>.
- Halvorsen, H.-P. (2014). Chapter 18. state estimation with kalman filter. URL [http://home.hit.no/~hansha/documents/control/theory/stateestimation\\_with\\_kalmanfilter.pdf](http://home.hit.no/~hansha/documents/control/theory/stateestimation_with_kalmanfilter.pdf). Accessed on January 2014.
- Hang, C., Åström, K., and Ho, W. (1991). Refinements of the Ziegler-Nichols tuning formula. *Control Theory and Applications, IEE Proceedings D*, 138(2):111 –118.
- Härdle, W. (1990). *Applied Nonparametric Regression*. Econometric Society Monographs. Cambridge University Press.
- 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.
- Haventon, M. and Öberg, J. (2008). Testing and implementation of a backlash detection algorithm. Master's thesis, Lund University, Sweden.
- Haykin, S. (1998). *Neural Networks: A Comprehensive Foundation*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2nd edition.
- He, Q. and Wang, J. (2010). Valve stiction modeling: First-principles vs data-driven approaches. *Proceedings of the American Control Conference*, pages 3777–3782. URL [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=5531561&tag=1](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5531561&tag=1).
- He, Q. P. and Wang, J. (2014). A valve stiction quantification method based on a semi-physical valve stiction model. *Industrial & Engineering Chemistry Research*, 53(30):12010–12022. URL <http://pubs.acs.org/doi/abs/10.1021/ie501069n>.
- He, Q. P., Wang, J., Pottmann, M., and Qin, S. J. (2007). A curve fitting method for detecting valve stiction in oscillating control loops. *Ind. Eng. Chem. Res.*, 46(13):4549–4560.
- He, X. B. and Yang, Y. P. (2008). Variable mwpc for adaptive process monitoring. *Industrial & Engineering Chemistry Research*, 47(2):419–427.
- Hensen, R. (2002). *Controlled mechanical systems with friction*. PhD thesis, Eindhoven University of Technology, Eindhoven (The Netherlands). URL <http://alexandria.tue.nl/extra2/200210527.pdf>.
- Hensen, R., Van de Molengraft, M. J. G., and Steinbuch, M. (2002). Frequency domain identification of dynamic friction model parameters. *IEEE Transactions on Control Systems Technology*, 10(2):191–196. URL <http://dx.doi.org/10.1109/87.987064>.
- Hindmarsh, A. C. (1983). ODEPACK, a systematized collection of ODE solvers. *IMACS Transactions on Scientific Computation*, 1:55–64.
- Hirschorn, R. and Miller, G. (1999). Control of nonlinear systems with friction. *IEEE Transactions on Control Systems Technology*, 7(5):588–595. URL <http://dx.doi.org/10.1109/87.784422>.
- Hof, P. M. V. D. and Schrama, R. J. (1995). Identification and control — closed-loop issues. *Automatica*, 31(12):1751 – 1770. Trends in System Identification.

## Bibliography

---

- Honeywell (2012). Loop Scout. URL <http://www.loopscout.com/>. Accessed on September 2012.
- Horch, A. (1999). A simple method for oscillation diagnosis in process control loops. In *Control Applications, 1999. Proceedings of the 1999 IEEE International Conference on*, volume 2, pages 1284–1289 vol. 2.
- Horch, A. (2000). *Condition Monitoring of Control Loops*. PhD thesis, Royal Institute of Technology, Stockholm (Sweden).
- Horch, A., Cox, J., and Bonavita, N. (2007). Peak performance: Root cause analysis of plant-wide disturbances. ABB.
- Horton, E. C., Foley, M. W., and Kwok, K. E. (2003). Performance assessment of level controllers. *International Journal of Adaptive Control and Signal Processing*, 17(7-9):663–684.
- Hovd, M. and Skogestad, S. (1993). Improved independent design of robust decentralized controllers. *J. Process Control*, 3:43 – 51.
- Hovd, M. and Skogestad, S. (1994). Sequential design of decentralized controllers. *Automatica*, 30(10):1601 – 1607.
- Howard, R. and Cooper, D. (2010). A novel pattern-based approach for diagnostic controller performance monitoring. *Control Engineering Practice*, 18(3):279 – 288.
- Huang, B. (1999). Performance assessment of processes with abrupt changes of disturbances. *The Canadian Journal of Chemical Engineering*, 77(5):1044–1054.
- Huang, B. (2002). Minimum variance control and performance assessment of time-variant processes. *Journal of Process Control*, 12(6):707 – 719.
- Huang, B. (2003). A pragmatic approach towards assessment of control loop performance. *International Journal of Adaptive Control and Signal Processing*, 17(7-9):589–608.
- Huang, B. (2011). Improve measurement with soft sensors for process industries. Presentation at ISA Fort McMurray. URL <http://isaedmontonshow.ca/old.isaedmonton.org/wp-content/uploads/Soft-Sensor-ISA-Pub.pdf>.
- Huang, B., Ding, S. X., and Qin, S. (2005). Closed-loop subspace identification: an orthogonal projection approach. *Journal of Process Control*, 15(1):53 – 66.

- Huang, B., Ding, S. X., and Thornhill, N. (2006). Alternative solutions to multi-variate control performance assessment problems. *Journal of Process Control*, 16(5):457 – 471.
- Huang, B. and Kadali, R. (2008). *Dynamic Modeling, Predictive Control and Performance Monitoring: A Data-Driven Subspace Approach*. Lecture Notes in Control and Information Sciences. Springer.
- Huang, B. and Shah, S. (1999). *Performance Assessment of Control Loops: Theory and Applications*. Advances in Industrial Control. Springer.
- Huang, B., Shah, S., and Miller, R. (2000a). 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. ADCHEM 1997 IFAC Symposium: Advanced Control of Chemical Processes.
- Huang, S., Tan, K., and Lee, T. (2000b). Adaptive friction compensation using neural network approximations. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 30(4):551–557. URL <http://dx.doi.org/10.1109/5326.897081>.
- Huba, M., Skogestad, S., Fikar, M., Hovd, M., Johnsen, T., and Rohal-Ilkiv, B. (2011). *Selected Topics on Constrained and Nonlinear Control*. Slovak University of Technology in Bratislava.
- Hugo, A. J. (2006). Performance assessment of single-loop industrial controllers. *Journal of Process Control*, 16(8):785 – 794.
- Hägglund, T. (1995). A Control-Loop Performance Monitor. *Control Engineering Practice*, 3:1543–1551.
- Hägglund, T. (2002). A friction compensator for pneumatic control valves. *Journal of Process Control*, 12(8):897 – 904.
- Hägglund, T. (2007). Automatic on-line estimation of backlash in control loops. *Journal of Process Control*, 17(6):489 – 499.
- Ikonen, E. and Najim, K. (2001). *Advanced Process Identification and Control*. Automation and Control Engineering. Taylor & Francis.

## Bibliography

---

- Ingimundarson, A. (2003). *Dead-Time Compensation and Performance Monitoring in Process Control*. PhD thesis, Department of Automatic Control, Lund University, Sweden.
- Ingimundarson, A. (2006). Performance assessment using a normalized gradient. *Journal of Process Control*, 16(10):1013 – 1020.
- IPS Engineering (2014). Oleochemicals: Glycerin recovery processing plant. URL <http://www.ips-engineering.it/Doc/Glycerin%20Recovery.pdf>, accessed on July 2015.
- ISA (1995). *Process Instrumentation Terminology*. URL [http://www.isa.org/Template.cfm?Section=Advanced\\_Search&Template=/Ecommerce/ProductDisplay.cfm&ProductID=2523](http://www.isa.org/Template.cfm?Section=Advanced_Search&Template=/Ecommerce/ProductDisplay.cfm&ProductID=2523).
- Ivan, L. Z. X. and Lakshminarayanan, S. (2009). A new unified approach to valve stiction quantification and compensation. *Ind. Eng. Chem. Res.*, 48(7):3474–3483.
- Jackson, J. (2003). *A User's Guide to Principal Components*. Wiley Series in Probability and Statistics. Wiley.
- Jacobs, R. A., Jordan, M. I., Nowlan, S. J., and Hinton, G. E. (1991). Adaptive mixtures of local experts. *Neural Comput.*, 3(1):79–87.
- Jamaludin, Z., Brussel, H. V., and Swevers, J. (2009). Design of a disturbance observer and model-based friction feedforward to compensate quadrant glitches. *Proceedings of the International Conference on Motion and Vibration Control*, pages 143–154. URL [http://dx.doi.org/10.1007/978-1-4020-9438-5\\_15](http://dx.doi.org/10.1007/978-1-4020-9438-5_15).
- Jang, J., Sun, C., and Mizutani, E. (1997). *Neuro-fuzzy and soft computing: a computational approach to learning and machine intelligence*. MATLAB curriculum series. Prentice Hall.
- Jang, J. O., Chung, H. T., and Jeon, G. J. (2005). Saturation and deadzone compensation of systems using neural network and fuzzy logic. In *American Control Conference, 2005. Proceedings of the 2005*, pages 1715–1720 vol. 3.
- Jelali, M. (2006). An overview of control performance assessment technology and industrial applications. *Control Engineering Practice*, 14(5):441 – 466. Intelligent Control Systems and Signal Processing ICONS 2003, IFAC International Conference on Intelligent Control Systems and Signal Processing.

- Jelali, M. (2008). Estimation of valve stiction in control loops using separable least-squares and global search algorithms. *J. Process Contr.*, 18(7-8):632–642.
- 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.
- Jelali, M. and Huang, B. (2010). *Detection and Diagnosis of Stiction in Control Loops: State of the Art and Advanced Methods*. Springer.
- Jelali, M. and Huang, B. (2013). Book homepage: Detection and diagnosis of stiction in control loops: State of the art and advanced methods. URL <http://www.ualberta.ca/~bhuang/Stiction-Book/>. Accessed on August 2013.
- Jiang, H., Choudhury, M., Shah, S., Cox, J., and Paulinis, M. (2006a). Detection and diagnosis of plant-wide oscillations via the method of spectral envelope. In *Proceedings of the IFAC-ACHEM, Gramado (Brazil)*.
- Jiang, H., Choudhury, M., Shah, S., Cox, J., and Paulinis, M. (2006b). Detection and diagnosis of plant-wide oscillations via the method of spectral envelope. In *Proceedings of the IFAC-ACHEM, Gramado (Brazil)*.
- Jianxu, L. and Huihe, S. (2002). Soft sensing modeling using neurofuzzy system based on rough set theory. In *American Control Conference, 2002. Proceedings of the 2002*, volume 1, pages 543–548 vol.1.
- Joachims, T. (2005). A support vector method for multivariate performance measures. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 377–384. ACM Press.
- Johnson, M. and Moradi, M. (2006). *PID Control: New Identification and Design Methods*. Probability and its applications. Springer.
- Jolliffe, I. (2002). *Principal Component Analysis*. Springer Series in Statistics. Springer.
- Jordaan, E., Kordon, A., and Chiang, L. (2006). Robust soft sensors based on ensemble of symbolic regression-based predictors. In Di Bucchianico, A., Mattheij, R., and Peletier, M., editors, *Progress in Industrial Mathematics at ECMI 2004*, volume 8 of *Mathematics in Industry*, pages 600–604. Springer Berlin Heidelberg.

## Bibliography

---

- Jordaan, E., Kordon, A., Chiang, L., and Smits, G. (2004). Robust inferential sensors based on ensemble of predictors generated by genetic programming. In Yao, X., Burke, E., Lozano, J., Smith, J., Merelo-Guervós, J., Bullinaria, J., Rowe, J., Tiño, P., Kabán, A., and Schwefel, H.-P., editors, *Parallel Problem Solving from Nature - PPSN VIII*, volume 3242 of *Lecture Notes in Computer Science*, pages 522–531. Springer Berlin Heidelberg.
- 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.
- Kadlec, P., Grbic, R., and Gabrys, B. (2011). Review of adaptation mechanisms for data-driven soft sensors. *Computers & Chemical Engineering*, 35(1):1 – 24.
- Kadlec, P. (2009). *On robust and adaptive soft sensors*. PhD thesis, Bournemouth University, United Kingdom.
- Kadlec, P. and Gabrys, B. (2007). Nature-inspired adaptive architecture for soft sensor modelling. In *3rd European Symposium on Nature-inspired Smart Information Systems, NiSIS'2007 Symposium*, St Julian's (Malta).
- Kadlec, P. and Gabrys, B. (2008a). Adaptive local learning soft sensor for inferential control support. In *Computational Intelligence for Modelling Control Automation, 2008 International Conference on*, pages 243–248.
- Kadlec, P. and Gabrys, B. (2008b). Learnt topology gating artificial neural networks. In *Neural Networks, 2008. IJCNN 2008. (IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on*, pages 2604–2611.
- Kadlec, P. and Gabrys, B. (2009). Architecture for development of adaptive on-line prediction models. *Memetic Computing*, 1(4):241–269.
- Kadlec, P. and Gabrys, B. (2010). Adaptive on-line prediction soft sensing without historical data. In *The 2010 International Joint Conference on Neural Networks (IJCNN)*, pages 1 – 8, Barcelona.
- Kadlec, P. and Gabrys, B. (2011). Local learning-based adaptive soft sensor for catalyst activation prediction. *AIChE Journal*, 57(5):1288–1301.
- Kadlec, P., Gabrys, B., and Strandt, S. (2009). Data-driven soft sensors in the process industry. *Computers & Chemical Engineering*, 33(4):795 – 814.

- Kalaivani, S., Aravind, T., and Yuvaraj, D. (2014). A single curve piecewise fitting method for detecting valve stiction and quantification in oscillating control loops. In Babu, B. V., Nagar, A., Deep, K., Pant, M., Bansal, J. C., Ray, K., and Gupta, U., editors, *Proceedings of the International Conference on Soft Computing for Problem Solving*, volume 236 of *Advances in Intelligent Systems and Computing*, pages 13–24. Springer India. URL [http://dx.doi.org/10.1007/978-81-322-1602-5\\_2](http://dx.doi.org/10.1007/978-81-322-1602-5_2).
- Kammer, L., Bitmead, R., and Bartlett, P. (1996). Signal-based testing of lq-optimality of controllers. In *Decision and Control, 1996., Proceedings of the 35th IEEE*, volume 4, pages 3620–3624 vol.4.
- Kaneko, H., Arakawa, M., and Funatsu, K. (2008). Development of a new regression analysis method using independent component analysis. *Journal of Chemical Information and Modeling*, 48(3):534–541. PMID: 18321087.
- Kaneko, H. and Funatsu, K. (2013). Adaptive soft sensor model using online support vector regression with time variable and discussion of appropriate parameter settings. *Procedia Computer Science*, 22(0):580–589. 17th International Conference in Knowledge Based and Intelligent Information and Engineering Systems - {KES2013}.
- Kaneko, H. and Funatsu, K. (2014). Adaptive soft sensor based on online support vector regression and bayesian ensemble learning for various states in chemical plants. *Chemometrics and Intelligent Laboratory Systems*, 137(0):57–66.
- Kano, M. and Fujiwara, K. (2013). Virtual sensing technology in process industries: Trends and challenges revealed by recent industrial applications. *JOURNAL OF CHEMICAL ENGINEERING OF JAPAN*, 46(1):1–17.
- Kano, M., Lee, S., and Hasebe, S. (2009). Two-stage subspace identification for softsensor design and disturbance estimation. *Journal of Process Control*, 19(2):179–186.
- Kano, M., Maruta, H., Kugemoto, H., and Shimizu, K. (2004). Practical model and detection algorithm for valve stiction. In *IFAC Symposium on Dynamics and Control of Process Systems*.
- Karnopp, D. (1985). Computer simulation of stick-slip friction in mechanical dynamic systems. *Journal of Dynamic Systems, Measurements and Controls*, 107(1):100–103. URL <http://dx.doi.org/10.1115/1.3140698>.

## Bibliography

---

- Karra, S. and Karim, M. N. (2009). Comprehensive methodology for detection and diagnosis of oscillatory control loops. *Control Eng. Pract.*, 17(8):939–956.
- Karthiga, D. and Kalaivani, S. (2012). A new stiction compensation method in pneumatic control valves. *Int. J. Electron. Comput. Sci. Eng.*, pages 2604–2612.
- Kato, T. and Hatanaka, K. (1998). Backlash compensation method for semi-closed-loop type servo control. US Patent 5,742,144.
- Kayihan, A. and Doyle-III, F. J. (2000). Friction compensation for a process control valve. *Control Engineering Practice*, 8(7):799 – 812.
- Khatibisepehr, S. (2013). *Bayesian solutions to multi-model inferential sensing problems*. PhD thesis, University of Alberta, Alberta, Canada.
- Khatibisepehr, S. and Huang, B. (2008). Dealing with irregular data in soft sensors: Bayesian method and comparative study. *Industrial & Engineering Chemistry Research*, 47(22):8713–8723.
- Khatibisepehr, S., Huang, B., and Khare, S. (2013). Design of inferential sensors in the process industry: A review of bayesian methods. *Journal of Process Control*, 23(10):1575 – 1596. URL <http://www.sciencedirect.com/science/article/pii/S095915241300156X>.
- Kim, J. and Shao, J. (2013). *Statistical methods for handling incomplete data*. CRC Press, Boca Raton.
- Kim, S., Kano, M., Hasebe, S., Takinami, A., and Seki, T. (2013). Long-term industrial applications of inferential control based on just-in-time soft-sensors: Economical impact and challenges. *Industrial & Engineering Chemistry Research*, 52(35):12346–12356.
- King, M. (2011). *Process Control: A Practical Approach*. John Wiley & Sons.
- Klein, V. and Morelli, E. (2006). *Aircraft system identification: theory and practice*. AIAA Education Series. American Institute of Aeronautics and Astronautics.
- 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. URL <http://www.sciencedirect.com/science/article/pii/S0005109801000322>.
- Ko, B.-S. and Edgar, T. F. (2004). Pid control performance assessment: The single-loop case. *AIChE Journal*, 50(6):1211–1218.
- Komulainen, T., Sourander, M., and Jämsä-Jounela, S.-L. (2004). An online application of dynamic {PLS} to a dearomatization process. *Computers & Chemical Engineering*, 28(12):2611 – 2619.
- Kordon, A., Jordaan, E., Chew, L., Smits, G., Bruck, T., Haney, K., and Jenings, A. (2004). Biomass inferential sensor based on ensemble of models generated by genetic programming. In Deb, K., editor, *Genetic and Evolutionary Computation – GECCO 2004*, volume 3103 of *Lecture Notes in Computer Science*, pages 1078–1089. Springer Berlin Heidelberg.
- Kosina, P., Gama, J. a., and Sebastião, R. (2010). Drift severity metric. In *Proceedings of the 2010 Conference on ECAI 2010: 19th European Conference on Artificial Intelligence*, pages 1119–1120, Amsterdam, The Netherlands, The Netherlands. IOS Press.
- Kozub, D. J. (2002). Controller performance monitoring and diagnosis: Industrial perspective. In *Proceedings of the 15th IFAC World Congress, 2002*, pages 1619 – 1619, Barcelona, Spain.
- Kuehl, P. and Horch, A. (2005). Detection of sluggish control loops-experiences and improvements. *Control Engineering Practice*, 13(8):1019 – 1025. Fault Detection, Supervision and Safety of Technical Processes (Safeprocess 2003).
- Kuncheva, L. (2004). Classifier ensembles for changing environments. In Roli, F., Kittler, J., and Windeatt, T., editors, *Multiple Classifier Systems*, volume 3077 of *Lecture Notes in Computer Science*, pages 1–15. Springer Berlin Heidelberg.
- Kuncheva, L. I. (2008). Classifier ensembles for detecting concept change in streaming data: Overview and perspectives. In *2nd Workshop SUEMA 2008 (ECAI 2008)*, pages 5–10.

## Bibliography

---

- Kuncheva, L. I. and Žliobaitė, I. (2009). On the window size for classification in changing environments. *Intell. Data Anal.*, 13(6):861–872.
- Kvam, A. (2009). Detection of stiction in control valves – An algorithm for the offshore oil and gas industry. Master’s thesis, Norwegian University of Science and Technology, Trondheim (Norway). URL <http://ntnu.diva-portal.org/smash/record.jsf?pid=diva2:347745>.
- Lahiri, S. (2003). *Resampling Methods for Dependent Data*. Springer Series in Statistics. Springer.
- Lampaert, V., Al-Bender, F., and Swevers, J. (2003). A generalized Maxwell-Slip friction model appropriate for control purposes. *Proceedings of the International Conference on Physics and Control*, 4:1170–1177. URL <http://dx.doi.org/10.1109/PHYCON.2003.1237071>.
- Lampaert, V., Swevers, J., and Al-Bender, F. (2002). Modification of the Leuven integrated friction model structure. *IEEE Transactions on Automatic Control*, 47(4):683–687. URL <http://dx.doi.org/10.1109/9.995050>.
- Lampaert, V., Swevers, J., and Al-Bender, F. (2004). Comparison of model and non-model based friction compensation techniques in the neighbourhood of pre-sliding friction. *Proceedings of the American Control Conference*, 2:1121–1126. URL [http://ieeexplore.ieee.org/xpl/freeabs\\_all.jsp?arnumber=1386722&abstractAccess=no&userType=inst](http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1386722&abstractAccess=no&userType=inst).
- Lanfredi, R. A. B. (2011). Comparação de métodos de compensação de agarramento em válvulas de controle. Master’s thesis, Federal University of Rio Grande do Sul, Rio Grande do Sul (Brazil). URL <http://www.lume.ufrgs.br/bitstream/handle/10183/38509/000823878.pdf?sequence=1>.
- Latwesen, A. L. and Junk, K. W. (2002). Statistical determination of estimates of process control loop parameters. Patent US 6466893. URL:[http://www.misclens.net/misclens/misc/US\\_6466893/en/](http://www.misclens.net/misclens/misc/US_6466893/en/).
- Lee, K. H., Ren, Z., and Huang, B. (2008). Novel closed-loop stiction detection and quantification method via system identification. In *International Symposium on Advanced Control of Industrial Processes*, Alberta (Canada).

- Lee, K. H., Tamayo, E. C., and Huang, B. (2010). Industrial implementation of controller performance analysis technology. *Control Eng. Pract.*, 18(2):147–158.
- Lee, Y., Park, S., Lee, M., and Brosilow, C. (1998). PID controller tuning for desired closed-loop responses for si/so systems. *AIChE Journal*, 44(1):106–115.
- Lei, F., Jiandong, W., and Qinghua, Z. (2013). Identification of extended hammerstein systems for modeling sticky control valves under general types of input signals. pages 1698–1703. URL [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=6639700](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6639700).
- Leine, R., van Campen, D., de Kraker, A., and van den Steen, L. (1998). Stick-slip vibrations induced by alternate friction models. *Nonlinear Dynamics*, 16:41–54. URL <http://dx.doi.org/10.1023/A%3A1008289604683>.
- Leon, D. D. (2012). Model fitting. Technical report, California State University, Fresno. Consulted in June 2012.
- Leonard, N. E. and Krishnaprasad, P. S. (1992). Adaptive friction compensation for bi-directional low velocity position tracking. Technical report, University of Maryland, Maryland (USA). URL <http://www.dtic.mil/dtic/tr/fulltext/u2/a455018.pdf>.
- Levenberg, K. (1944). A method for the solution of certain non-linear problems in least squares. *Quarterly Journal of Applied Mathematics*, II(2):164–168.
- Li, H., Yu, D., and Braun, J. E. (2011). A review of virtual sensing technology and application in building systems. *HVAC&R Research*, 17(5):619–645.
- Li, Q., Du, Q., Ba, W., and Shao, C. (2012). Multiple-input multiple-output soft sensors based on KPCA and MKLS-SVM for quality prediction in atmospheric distillation column. *International Journal of Innovative Computing, Information and Control*, 8(12).
- Li, Q., Whiteley, J., and Rhinehart, R. (2003). A relative performance monitor for process controllers. *International Journal of Adaptive Control and Signal Processing*, 17(7-9):685–708.
- Li, W., Yue, H., Valle-Cervantes, S., and Qin, S. (2000). Recursive PCA for adaptive process monitoring. *Journal of Process Control*, 10(5):471 – 486.

## Bibliography

---

- Li, Y., Ang, K. H., and Chong, G. (2006). Patents, software, and hardware for PID control: an overview and analysis of the current art. *Control Systems, IEEE*, 26(1):42 – 54.
- Liang, Z., Chunming, X., Jiabin, C., and Jianrong, Z. (2012). Physical-based modeling of nonlinearities in process control valves. *Proceedings of the International Conference on Control Engineering and Communication Technology*, pages 75–78. URL <http://dx.doi.org/10.1109/ICCECT.2012.170>.
- Lin, B., Recke, B., Knudsen, J. K., and Jørgensen, S. B. (2007). A systematic approach for soft sensor development. *Computers & Chemical Engineering*, 31(5–6):419 – 425. ESCAPE-15 Selected Papers from the 15th European Symposium on Computer Aided Process Engineering held in Barcelona, Spain, May 29-June 1, 2005.
- Lin, B., Recke, B., Renaudat, P., Knudsen, J., and Jorgensen, S. (2005). Robust statistics for soft sensor development in cement kiln. In *Proceedings of the 16th IFAC World Congress, 2005*, volume 16:1.
- Ling, B., Zeifman, M., and Liu, M. (2007). A practical system for online diagnosis of control valve faults. In *Decision and Control, 2007 46th IEEE Conference on*, pages 2572–2577.
- Linhares, L. L. S. (2010). Sistema híbrido de inferência baseado em análise de componentes principais e redes neurais artificiais aplicado a plantas de processamento de gás natural. Master's thesis, Universidade Federal do Rio Grande do Norte, Brazil.
- Little, R. J. A. (1988). A test of missing completely at random for multivariate data with missing values. *Journal of the American Statistical Association*, 83(404):1198 – 1202.
- Liu, H., Huang, M., and Yoo, C. (2013). A fuzzy neural network-based soft sensor for modeling nutrient removal mechanism in a full-scale wastewater treatment system. *Desalination and Water Treatment*, 51(31-33):6184–6193.
- Liu, T. and Gao, F. (2011). *Industrial process identification and control design: step-test and relay-experiment-based methods*. Advances in Industrial Control. Springer.
- Liu, X., Kruger, U., Littler, T., Xie, L., and Wang, S. (2009). Moving window kernel {PCA} for adaptive monitoring of nonlinear processes. *Chemometrics and Intelligent Laboratory Systems*, 96(2):132 – 143. Chimie 2007, Lyon, France, 29-30 November 2007.

- Liu, Y., Huang, D., and Li, Y. (2012). Development of interval soft sensors using enhanced just-in-time learning and inductive confidence predictor. *Industrial & Engineering Chemistry Research*, 51(8):3356–3367.
- Ljung, L. (1999). *System identification: theory for the user*. Prentice-Hall information and system sciences series. Prentice-Hall.
- Lobo, L. Q. and Ferreira, A. G. M. (2006). *Termodinâmica e Propriedades Termofísicas – Volume I: Termodinâmica das Fases*. Imprensa da Universidade de Coimbra, Coimbra.
- Lopez, A., Miller, C., Smith, C., and Murrill, P. (1967). Controller tuning relationships based on integral performance criteria. *Instrumentation Technology*, 14(12):72.
- LPMOM (2013). Laboratory of Process Monitoring, Optimization and Management. URL <http://www.mech.pku.edu.cn/robot/teacher/wangjiandong/research.htm>. Accessed on August 2013.
- Lu, C.-J., Shao, Y. E., Li, P.-H., and Wang, Y.-C. (2010). Recognizing mixture control chart patterns with independent component analysis and support vector machine. In *Proceedings of the 7th international conference on Advances in Neural Networks - Volume Part II*, ISSN'10, pages 426–431, Berlin, Heidelberg. Springer-Verlag.
- Luttmann, R., Bracewell, D. G., Cornelissen, G., Gernaey, K. V., Glassey, J., Hass, V. C., Kaiser, C., Preusse, C., Striedner, G., and Mandenius, C.-F. (2012). Soft sensors in bioprocessing: A status report and recommendations. *Biotechnology Journal*, 7(8):1040–1048.
- Luyben, W. L. (1986). A simple method for tuning siso controllers in multivariable systems. *Ind. Eng. Chem. Proc. Des. Devel.*, 25:654 ff.
- Lynch, C. and Dumont, G. (1996). Control loop performance monitoring. *Control Systems Technology, IEEE Transactions on*, 4(2):185–192.
- Lyzell, C. (2009). Initialization methods for system identification. Master's thesis, Linköping University. URL <http://www.diva-portal.org/smash/get/diva2:277019/FULLTEXT02>.
- Mah, R., Tamhane, A., Tung, S., and Patel, A. (1995). Process trending with piecewise linear smoothing. *Computers & Chemical Engineering*, 19(2):129–137.

## Bibliography

---

- Majecki, P. and Grimble, M. J. (2004). Gmv and restricted - structure gmv controller performance assessment - multivariable case. In *Proceedings of the 2004 American Control Conference*, volume 1, pages 697 – 702.
- Makkar, C. (2006). *Nonlinear modeling, identification, and compensation for frictional disturbances*. PhD thesis, University of Florida, Florida (USA).
- Makkar, C., Dixon, W., Sawyer, W., and Hu, G. (2005). A new continuously differentiable friction model for control systems design. *Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pages 600–605. URL <http://dx.doi.org/10.1109/AIM.2005.1511048>.
- Manoj, K. and Senthamarai, K. (2013). Comparison of methods for detecting outliers. *International Journal of Scientific & Engineering Research*, 4(9).
- Manosak, R., Limpattayanate, S., and Hunsom, M. (2011). Sequential-refining of crude glycerol derived from waste used-oil methyl ester plant via a combined process of chemical and adsorption. *Fuel Processing Technology*, 92(1):92 – 99.
- Mansano, R. K., Godoy, E. P., and Porto, A. J. V. (2014). The benefits of soft sensor and multi-rate control for the implementation of wireless networked control systems. *Sensors*, 14(12):24441–24461.
- Manum, H. (2006). *Analysis of techniques for automatic detection and quantification of stiction in control loops*. PhD thesis, University of Pisa, Italy.
- Manum, H. and Scali, C. (2006). Closed loop performance monitoring: Automatic diagnosis of valve stiction by means of a technique based on shape analysis formalism. *Proceedings of the International Congress on Methodologies for Emerging Technologies in Automation*. URL [http://www.nt.ntnu.no/users/skoge/publications/2006/manum\\_scali\\_anipla06/manum-M211.pdf](http://www.nt.ntnu.no/users/skoge/publications/2006/manum_scali_anipla06/manum-M211.pdf).
- Maria, A. (1997). Introduction to modeling and simulation. In *Proceedings of the 29th Conference on Winter Simulation, WSC '97*, pages 7–13, Washington (USA). IEEE Computer Society.
- Márton, L. and Lantos, B. (2007). Modeling, identification, and compensation of stick-slip friction. *IEEE Transactions on Industrial Electronics*, 54(1):511–521. URL <http://dx.doi.org/10.1109/TIE.2006.888804>.

- Maruta, H., Kano, M., Kugemoto, H., and Shimizu, K. (2005). Modeling and detection of stiction in pneumatic control valves. *Transactions of the Society of Instrument and Control Engineers*, E4(1):18–26. URL <http://srv01.sice.or.jp/~e-trans/papers/E4-3.pdf>.
- Masson, M. H., Canu, S., Grandvalet, Y., and Lynggaard-Jensen, A. (1999). Software sensor design based on empirical data. *Ecological Modelling*, 120(2–3):131 – 139.
- Matrikon (2012). Matrikon Control Performance Monitor. URL <http://www.matrikon.com/control-performance-monitoring/control-performance-monitor.aspx>. Accessed on August 2012.
- Matsumoto, S., Kamei, Y., Monden, A., and Matsumoto, K.-i. (2007). Comparison of outlier detection methods in fault-proneness models. In *Empirical Software Engineering and Measurement, 2007. ESEM 2007. First International Symposium on*, pages 461–463.
- Matsuo, T., Sasaoka, H., and Yamashita, Y. (2003). Detection and diagnosis of oscillations in process plants. In Palade, V., Howlett, R., and Jain, L., editors, *Knowledge-Based Intelligent Information and Engineering Systems*, volume 2773 of *Lecture Notes in Computer Science*, pages 1258–1264. Springer Berlin / Heidelberg. 10.1007/978-3-540-45224-9\_170.
- 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.
- Mendes, J., Souza, F., Araujo, R., and Goncalves, N. (2012). Genetic fuzzy system for data-driven soft sensors design. *Applied Soft Computing*, 12(10):3237 – 3245.
- Meng, K., Li, S., and Yang, X. (2009). A robust SQP method based on a smoothing lower order penalty function. *Optimization*, 58(1):23–38.
- Merriam-Webster Dictionary (2012). Word of the day – stiction. URL <http://www.merriam-webster.com/word-of-the-day/2012/02/02/>. Accessed in May 2014.
- Meyer, P. E. (2008). *Information-theoretic variable selection and network inference from microarray data*. PhD thesis, Université Libre de Bruxelles, Bruxelles (Belgique). URL <http://theses.ulb.ac.be/ETD-db/collection/available/ULBetd-12162008-093634/unrestricted/meyerTheseDistrib.pdf>.

## Bibliography

---

- Miao, T. and Seborg, D. E. (1999). Automatic Detection of Excessively Oscillatory Feedback Control Loops. In *Proceedings of the 1999 IEEE International Conference on Control Applications*, volume 1, pages 359 – 365, Kohala Coast - Island of Hawai'i, USA.
- Minku, L., White, A., and Yao, X. (2010). The impact of diversity on online ensemble learning in the presence of concept drift. *Knowledge and Data Engineering, IEEE Transactions on*, 22(5):730–742.
- Mohammad, M. A. and Huang, B. (2012). Compensation of control valve stiction through controller tuning. *Journal of Process Control*, 22(9):1800 – 1819. URL <http://www.sciencedirect.com/science/article/pii/S095915241200193X>.
- Moreno-Torres, J. G., Raeder, T., Alaiz-Rodríguez, R., Chawla, N. V., and Herrera, F. (2012). A unifying view on dataset shift in classification. *Pattern Recognition*, 45(1):521 – 530.
- Nallasivam, U., Babji, S., and Rengaswamy, R. (2010). Stiction identification in nonlinear process control loops. *Computers & Chemical Engineering*, 34(11):1890 – 1898.
- National Instruments Corporation (2015). Labview 2013 system identification toolkit help: Validating models. URL [http://zone.ni.com/reference/en-XX/help/372458D-01/lvsysidconcepts/validating\\_models/](http://zone.ni.com/reference/en-XX/help/372458D-01/lvsysidconcepts/validating_models/). Accessed on January 2015.
- Nayfeh, A. and Mook, D. (1995). *Nonlinear Oscillations*. Wiley-Interscience publication. Wiley.
- Ni, W., Brown, S. D., and Man, R. (2014). A localized adaptive soft sensor for dynamic system modeling. *Chemical Engineering Science*, 111(0):350 – 363.
- Nielsen, J. K. (2009). Sinusoidal parameter estimation. Master's thesis, Aalborg University. URL [http://projekter.aau.dk/projekter/files/17632341/bayesian\\_pitch\\_estimation\\_pdflatex.pdf](http://projekter.aau.dk/projekter/files/17632341/bayesian_pitch_estimation_pdflatex.pdf).
- NIST and SEMATECH (2012). *e-Handbook of Statistical Methods*. NIST. URL: <http://www.itl.nist.gov/div898/handbook/>.
- Nocedal, J. and Wright, S. (1999). *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer New York.



- O'Connor, N. and O'Dwyer, A. (2004). Control loop performance assessment: A classification of methods. In *Irish Signals and Systems Conference, Belfast (United Kingdom)*, pages 530–535.
- O'Dwyer, A. (2003). PID compensation of time delayed processes 1998-2002: a survey. In *American Control Conference, Denver, Colorado, USA*, pages 1494–1499.
- O'Dwyer, A. (2006). *Handbook of PI and PID Controller Tuning Rules*. Imperial College Press.
- Olsson, D. (2011). Applications and implementation of kernel principal component analysis to specific data sets. Master's thesis, Royal Institute of Technology, Stockholm (Sweden).
- Olsson, H. (1996). *Control Systems with Friction*. PhD thesis, Lund Institute of Technology, Lund (Sweden). URL [www.control.lth.se/documents/1996/olsh96dis.pdf](http://www.control.lth.se/documents/1996/olsh96dis.pdf).
- Olsson, H., Åström, K. J., de Wit, C. C., Gafvert, M., and Lischinsky, P. (1998). Friction models and friction compensation. *European Journal of Control*, 4(3):176–195. URL [http://www.gipsa-lab.grenoble-inp.fr/~carlos.canudas-de-wit/publications/friction/dynamic\\_friction\\_EJC\\_98.pdf](http://www.gipsa-lab.grenoble-inp.fr/~carlos.canudas-de-wit/publications/friction/dynamic_friction_EJC_98.pdf).
- Ordys, A., Uduehi, D., Johnson, M., and Thornhill, N. (2007). *Process Control Performance Assessment: From Theory to Implementation*. Advances in Industrial Control. Springer.
- Orr, M. J. L. (1996). Introduction to radial basis function networks. URL [url = http://www.cc.gatech.edu/~isbell/tutorials/rbf-intro.pdf](http://www.cc.gatech.edu/~isbell/tutorials/rbf-intro.pdf), <http://www.cc.gatech.edu/~isbell/tutorials/rbf-intro.pdf>. Accessed on March 2014.
- Otten, G., Vries, T. J. A. D., Rankers, A. M., and Gaal, E. W. (1997). Linear motor motion control using a learning feedforward controller. *IEEE/ASME Transactions on Mechatronics*, 2:179–187. URL <http://dx.doi.org/10.1109/3516.622970>.
- Palmor, Z., Halevi, Y., and Krasney, N. (1995). Automatic tuning of decentralized PID controllers for TITO processes. *Automatica*, 31(7):1001 – 1010.
- Pan, Q. (2007). *System identification of constructed civil engineering structures and uncertainty*. PhD thesis, Drexel University, Pennsylvania (USA).

## Bibliography

---

- Panteley, E., Ortega, R., and Gäfvert, M. (1998). An adaptive friction compensator for global tracking in robot manipulators. *Systems & Control Letters*, 33(5):307–313. URL <http://www.sciencedirect.com/science/article/pii/S0167691197001138>.
- Paprican (2012). Process control – automatic control loop monitoring and diagnostics. URL [http://www.paprican.ca/wps/portal/paprican/Technologies\\_Available/Technologies\\_for\\_Licensing/Process\\_Control?lang=en](http://www.paprican.ca/wps/portal/paprican/Technologies_Available/Technologies_for_Licensing/Process_Control?lang=en). Accessed on September 2012.
- PAS (2012). Control Loop Optimization. URL <http://www.pas.com/Solutions/Control-Loop-Performance.aspx>. Accessed on September 2012.
- Patwardhan, R. (1999). *Studies in Synthesis and Analysis of Model Predictive Controllers*. PhD thesis, University of Alberta.
- Pearson, R. (2005). *Mining Imperfect Data: Dealing with Contamination and Incomplete Records*. SIAM e-books. Society for Industrial and Applied Mathematics (SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104).
- Penny, K. I. and Jolliffe, I. T. (2001). A comparison of multivariate outlier detection methods for clinical laboratory safety data. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 50(3):295–307.
- Peretzki, D., Isaksson, A. J., Bittencourt, A. C., and Forsman, K. (2011). Data mining of historic data for process identification. Technical report, Linköpings Universitet, Sweden. Report no. LiTH-ISY-R-3039. Consulted in January 2015.
- Perrier, M. and Roche, A. A. (1992). Towards mill-wide evaluation of control loop performance. In *Proceedings of the control systems, Whistler (Canada)*, pages 205–209.
- Pierri, F., Paviglianiti, G., Caccavale, F., and Mattei, M. (2008). Observer-based sensor fault detection and isolation for chemical batch reactors. *Engineering Applications of Artificial Intelligence*, 21(8):1204 – 1216.
- Prasad, V., Schley, M., Russo, L. P., and Bequette, B. W. (2002). Product property and production rate control of styrene polymerization. *Journal of Process Control*, 12(3):353 – 372.
- Qi, F. and Huang, B. (2011). Estimation of distribution function for control valve stiction estimation. *J. Process Contr.*, 21(8):1208–1216.

- Qin, S. J. (1998). Control performance monitoring – a review and assessment. *Computers & Chemical Engineering*, 23(2):173 – 186.
- Qin, S. J. (2006). An overview of subspace identification. *Computers & Chemical Engineering*, 30(10–12):1502 – 1513. Papers from Chemical Process Control {VII} {CPC} {VII} Seventh international conference in the Series.
- 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.
- Rabinowicz, E. (1951). The nature of the static and kinetic coefficients of friction. *Journal of Applied Physics*, 22(11):1373–1379. URL <http://dx.doi.org/10.1063/1.1699869>.
- Radhakrishnan, K. and Hindmarsh, A. C. (1993). Description and Use of LSODE, the Livermore Solver for Ordinary Differential Equations. Technical report.
- Rallo, R., Ferré-giné, J., and Giral, F. (2003). Best feature selection and data completion for the design of soft neural sensors. In *Proceedings of AIChE 2003, 2nd Topical Conference on Sensors*.
- Åström, K. J. (1998). Control of systems with friction. *Proceedings of the International Conference on Motion and Vibration Control*, pages 25–32.
- Ratcliff, R. (1993). Methods for dealing with reaction time outliers. *Psychological Bulletin*, 114(3):510–532.
- Rato, T. (2009). Supervisão de sistemas de controlo. Master’s thesis, University of Coimbra, Portugal.
- Rebouças, D. L. (2009). Sistema de inferência neural e processamento estatístico multivariável aplicado a indústria do petróleo. Master’s thesis, Universidade Federal do Rio Grande do Norte, Brazil.
- Rengaswamy, R., Hägglund, T., and Venkatasubramanian, V. (2001). A qualitative shape analysis formalism for monitoring control loop performance. *Engineering Applications of Artificial Intelligence*, 14(1):23–33. URL <http://www.sciencedirect.com/science/article/pii/S0952197600000518>.

## Bibliography

---

- Rhinehart, R. (1995). A watchdog for controller performance monitoring. In *American Control Conference, 1995. Proceedings of the*, volume 3, pages 2239–2240 vol.3, Seattle.
- Rissanen, J. (1978). Modeling by shortest data description. *Automatica*, 14(5):465–471. URL <http://www.sciencedirect.com/science/article/pii/0005109878900055>.
- Ro, P. I. and Hubbel, P. I. (1993). Model reference adaptive control of dual-mode micro/macro dynamics of ball screws for nanometer motion. *Journal of Dynamic Systems, Measurement, and Control*, 115(1):103–108. URL <http://link.aip.org/link/?JDS/115/103/1>.
- Rodríguez-Liñán, M. d. C. and Heath, W. (2012). Mpc for plants subject to saturation and deadzone, backlash stiction. In *4th IFAC Nonlinear Model Predictive Control Conference*, volume 4.
- Rogina, A., Šiško, I., Mohler, I., Ujević, Z., and Bolf, N. (2011). Soft sensor for continuous product quality estimation (in crude distillation unit). *Chemical Engineering Research and Design*, 89(10):2070–2077.
- Romanenko, A., Brásio, A. S., Fernandes, N. C., Portugal, A. A., and Santos, L. O. (2014). Monitorização e optimização do desempenho de controladores na presença de falhas nos elementos finais de controlo. URL [http://worldwide.espacenet.com/publicationDetails/biblio?DB=worldwide.espacenet.com&II=0&ND=3&adjacent=true&locale=en\\_EP&FT=D&date=20140630&CC=PT&NR=106716A&KC=A](http://worldwide.espacenet.com/publicationDetails/biblio?DB=worldwide.espacenet.com&II=0&ND=3&adjacent=true&locale=en_EP&FT=D&date=20140630&CC=PT&NR=106716A&KC=A). Patent Application PT 106716 (A).
- Romanenko, A., Brásio, A. S., Fernandes, N. C., Portugal, A. A., and Santos, L. O. (2015). Método e sistema de análise do desempenho de anéis de controlo industriais pela deteção e caracterização automáticas de oscilações múltiplas. URL [http://worldwide.espacenet.com/publicationDetails/biblio?DB=worldwide.espacenet.com&II=21&ND=3&adjacent=true&locale=en\\_EP&FT=D&date=20150318&CC=PT&NR=107168A&KC=A](http://worldwide.espacenet.com/publicationDetails/biblio?DB=worldwide.espacenet.com&II=21&ND=3&adjacent=true&locale=en_EP&FT=D&date=20150318&CC=PT&NR=107168A&KC=A). Patent Application PT 107168 (A).
- Romano, R. and Garcia, C. (2010). Valve friction quantification and nonlinear process model identification. *Proceedings of the International Symposium on Dynamics and*

- 
- Control of Process Systems*, pages 115–120. URL <http://dx.doi.org/10.3182/20100705-3-BE-2011.00020>.
- Romano, R. A. and Garcia, C. (2011). Valve friction and nonlinear process model closed-loop identification. *Journal of Process Control*, 21(4):667 – 677.
- Rosner, B. (1983). Percentage points for a generalized esd many-outlier procedure. *Technometrics*, 25(2):165–172.
- Rossi, M., Tangirala, A., Shah, S., and Scali, C. (2006). A data-base measure for interactions in multivariate systems. In *IFAC-ADCHEM 2006, Gramado (Brazil)*.
- Rubin, D. B. (1976). Inference and missing data. *Biometrika*, 63(3):581–592.
- Salahshoor, K., Karimi, I., Fazel, E. N., and Beitari, H. (2011). Practical design and implementation of a control loop performance assessment package in an industrial plant. In *Proceedings of the 30th Chinese Control Conference*, pages 5888 – 5893, Yantai, China.
- Salahshoor, K., Kordestani, M., and Khoshro, M. (2009). Design of online soft sensors based on combined adaptive pca and rbf neural networks. In *Computational Intelligence in Control and Automation, 2009. CICA 2009. IEEE Symposium on*, pages 89–95.
- Salimbahrami, B. and Lohmann, B. (2006). Order reduction of large scale second-order systems using Krylov subspace methods. *Linear Algebra and its Applications*, 415(2-3):385–405.
- Salsbury, T. I. (2005). A practical method for assessing the performance of control loops subject to random load changes. *Journal of Process Control*, 15(4):393 – 405.
- Salsbury, T. I. (2006). Control performance assessment for building automation systems. In *IFAC Workshop on Energy Saving Control in Plants and Buildings, Bulgaria*, volume 1.
- Saporo, A. (2014). State of the art in the development of adaptive soft sensors based on just-in-time models. *Procedia Chemistry*, 9(0):226 – 234. International Conference and Workshop on Chemical Engineering {UNPAR} 2013 (ICCE {UNPAR} 2013).
- Schafer, J. (1997). *Analysis of Incomplete Multivariate Data*. Chapman & Hall/CRC Monographs on Statistics & Applied Probability. Taylor & Francis.
- Schafer, J. and Cinar, A. (2004). Multivariable mpc system performance assessment, monitoring, and diagnosis. *Journal of Process Control*, 14(2):113 – 129.

## Bibliography

---

- Schapire, R. E. (1990). The strength of weak learnability. In *Machine Learning*.
- Schmid, M. D. (2009). A neural network package for Octave – User’s Guide. URL [http://www.plexso.com/61\\_octave/neuralNetworkPackageForOctaveDevelop.pdf](http://www.plexso.com/61_octave/neuralNetworkPackageForOctaveDevelop.pdf). Accessed in March 2013.
- Schölkopf, B., Smola, A., and Müller, K.-R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Comput.*, 10(5):1299–1319.
- Schork, J. (1993). *Control of Polymerization Reactors*. Taylor & Francis.
- Schuster, M. and Paliwal, K. (1997). Bidirectional recurrent neural networks. *Trans. Sig. Proc.*, 45(11):2673–2681.
- Seborg, D., Edgar, T., Mellichamp, D., and Francis J. Doyle, I. (2010). *Process dynamics and control*. John Wiley & Sons.
- Seo, S. (2006). A review and comparison of methods for detecting outliers in univariate data sets. Master’s thesis, University of Pittsburgh, Pittsburgh (USA).
- Sepehri, N., Sassani, F., Lawrence, P. D., and Ghasempoor, A. (1996). Simulation and experimental studies of gear backlash and stick-slip friction in hydraulic excavator swing motion. *Journal of Dynamic Systems, Measurement and Control*, 118(3):463–467. URL <http://dx.doi.org/10.1115/1.2801168>.
- Seppala, C., Harris, T., and Bacon, D. (2002). Time series methods for dynamic analysis of multiple controlled variables. *Journal of Process Control*, 12(2):257 – 276.
- Shang, C., Huang, X., Suykens, J. A., and Huang, D. (2015). Enhancing dynamic soft sensors based on dpls: A temporal smoothness regularization approach. *Journal of Process Control*, 28(0):17 – 26.
- Shang, C., Yang, F., Huang, D., and Lyu, W. (2014). Data-driven soft sensor development based on deep learning technique. *Journal of Process Control*, 24(3):223 – 233.
- Shang, Q. L., Li, J. J., and Yu, S. E. (2013). Stiction characteristics identification of pneumatic control valve. *Information Technology Journal*, 12(18):4790–4796. URL <http://http://scialert.net/abstract/?doi=itj.2013.4790.4796>.
- Shao, W., Tian, X., and Chen, H. (2013). Adaptive anti-over-fitting soft sensing method based on local learning. In *10th IFAC International Symposium on Dynamics and Control of Process Systems*, pages 415–420.

- Shardt, Y. (2014). Process assessment technologies and solutions – developed software. URL <http://www.ualberta.ca/~control/programmes.html>. Accessed in July 2014.
- Shardt, Y., Zhao, Y., Qi, F., Lee, K., Yu, X., Huang, B., and Shah, S. (2012). Determining the state of a process control system: Current trends and future challenges. *The Canadian Journal of Chemical Engineering*, 90(2):217–245.
- Shardt, Y. A. and Huang, B. (2012). Design of experiments for nonlinear dynamic system identification. In *Proceedings of the Chemical Process Control VIII Conference, Georgia, United States of America*. URL <http://www.nt.ntnu.no/users/skoge/prost/proceedings/cpc8-focapo-2012/data/papers/076.pdf>.
- Shlens, J. (2009). A tutorial on principal component analysis. URL <http://www.sn1.salk.edu/~shlens/pca.pdf>. Accessed on September 2013.
- Shumway, R. and Stoffer, D. (2000). *Time Series Analysis and Its Applications*. Springer Texts In Statistics. Springer.
- Silva, B. C. and Garcia, C. (2014). Comparison of stiction compensation methods applied to control valves. *Industrial & Engineering Chemistry Research*, 53(10):3974–3984.
- Singhal, A. and Salsbury, T. I. (2005). A simple method for detecting valve stiction in oscillating control loops. *Journal of Process Control*, 15(4):371 – 382.
- Sivagamasundari, S. and Sivakumar, D. (2013). A new methodology to compensate stiction in pneumatic control valves. *International Journal of Soft Computing and Engineering*, 2(6):480–484. URL <http://www.ijcsce.org/attachments/File/v2i6/F1203112612.pdf>.
- Sjöberg, J. (2005). Neural Networks – Train and analyze neural networks to fit your data. Technical report, Wolfram Research. URL <http://media.wolfram.com/documents/NeuralNetworksDocumentation.pdf>. Accessed in March 2013.
- Slišković, D., Grbić, R., and Hocenski, Z. (2011). Methods for plant data-based process modeling in soft-sensor development. *Automatika*, 52(4):306–318.
- Smith, C. (1972). *Digital computer process control*. The Intext series in chemical engineering. Intext Educational Publishers.

## Bibliography

---

- Smith, L. I. (2002). A tutorial on principal component analysis. Technical report, Cornell University, Ithaca (USA).
- Soares, S., Araujo, R., Sousa, P., and Souza, F. (2011). Design and application of soft sensor using ensemble methods. In *Emerging Technologies Factory Automation (ETFA), 2011 IEEE 16th Conference on*, pages 1–8.
- Söderström, T. and Stoica, P. (1989). *System identification*. Prentice-Hall international series in systems and control engineering. Prentice Hall.
- Somieski, G. (2001). An eigenvalue method for calculation of stability and limit cycles in nonlinear systems. *Nonlinear Dynamics*, 26:3–22. URL <http://dx.doi.org/10.1023/A:3A1017384211491>.
- Soons, Z. I., Streefland, M., van Straten, G., and van Boxtel, A. J. (2008). Assessment of near infrared and “software sensor” for biomass monitoring and control. *Chemometrics and Intelligent Laboratory Systems*, 94(2):166 – 174.
- Sotomayor, O. A., Park, S. W., and Garcia, C. (2003). Multivariable identification of an activated sludge process with subspace-based algorithms. *Control Engineering Practice*, 11(8):961 – 969. *Process Dynamics and Control*.
- Souza, F. and Araujo, R. (2011). Variable and time-lag selection using empirical data. In *Emerging Technologies Factory Automation (ETFA), 2011 IEEE 16th Conference on*, pages 1–8. URL <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6059083>.
- Souza, F., Santos, P., and Araujo, R. (2010). Variable and delay selection using neural networks and mutual information for data-driven soft sensors. In *Emerging Technologies and Factory Automation (ETFA), 2010 IEEE Conference on*, pages 1–8.
- Spinelli, W., Piroddi, L., and Lovera, M. (2005). On the role of prefiltering in nonlinear system identification. *Automatic Control, IEEE Transactions on*, 50(10):1597 – 1602.
- Srinivasan, B., Spinner, T., and Rengaswamy, R. (2012). A reliability measure for model based stiction detection approaches. In *Symposium on Advanced Control of Chemical Processes*, pages 750–755.
- Srinivasan, R. and Rengaswamy, R. (2005). Stiction compensation in process control loops: A framework for integrating stiction measure and compensation. *Industrial*



- & *Engineering Chemistry Research*, 44(24):9164–9174. URL <http://pubs.acs.org/doi/abs/10.1021/ie050748w>.
- Srinivasan, R. and Rengaswamy, R. (2006a). Integrating stiction diagnosis and stiction compensation in process control valves. *Computer Aided Chemical Engineering*, 21B:1233–1238. URL <http://www.nt.ntnu.no/users/skoge/prost/proceedings/escape16-pse2006/Part%20B/Volume%2021B/N52257-Topic3/Topic%203-%20Oral/1310.pdf>.
- Srinivasan, R. and Rengaswamy, R. (2006b). Techniques for stiction diagnosis and compensation in process control loops. In *American Control Conference, 2006*, page 6 pp.
- Srinivasan, R. and Rengaswamy, R. (2007). Apparatus and method for stiction compensation in a process control system. US Patent 2007/0088446 A1.
- Srinivasan, R. and Rengaswamy, R. (2008). Approaches for efficient stiction compensation in process control valves. *Computers & Chemical Engineering*, 32(1–2):218 – 229. <ce:title>Process Systems Engineering: Contributions on the State-of-the-Art</ce:title> <ce:subtitle>Selected extended Papers from ESCAPE '16/PSE 2006.</ce:subtitle>.
- Srinivasan, R. and Rengaswamy, R. (2010). Apparatus and method for stiction compensation in a process control system. US Patent 7,797,082. URL <http://www.google.com/patents/US7797082>.
- Srinivasan, R., Rengaswamy, R., and Miller, R. (2005a). Control loop performance assessment. 1. a qualitative approach for stiction diagnosis. *Industrial & Engineering Chemistry Research*, 44(17):6708–6718.
- Srinivasan, R., Rengaswamy, R., and Miller, R. (2007). A modified empirical mode decomposition (EMD) process for oscillation characterization in control loops. *Control Engineering Practice*, 15(9):1135 – 1148.
- Srinivasan, R., Rengaswamy, R., Narasimhan, S., and Miller, R. (2005b). Control loop performance assessment. 2. Hammerstein model approach for stiction diagnosis. *Ind. Eng. Chem. Res.*, 44(17):6719–6728.
- 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.

## Bibliography

---

- Stefansky, W. (1972). Rejecting outliers in factorial designs. *Technometrics*, 14(2):469–479.
- Stenman, A., Gustafsson, F., and Forsman, K. (2003). A segmentation-based method for detection of stiction in control valves. *Int. J. Adapt. Control*, 17(7-9):625–634.
- Stewart, J. (2010). *Calculus: Early Transcendentals*. Textbooks Available with Cengage Youbook. Cengage Learning.
- Su, S. W., Nguyen, H., Jarman, R., Zhu, J., Lowe, D., McLean, P., Huang, S., Nguyen, N. T., Nicholson, R., and Weng, K. (2009). 3(2):837 – 841.
- Sumana, C. and Venkateswarlu, C. (2010). Genetically tuned decentralized proportional-integral controllers for composition control of reactive distillation. *Industrial & Engineering Chemistry Research*, 49(3):1297–1311.
- Sun, R. and Tsung, F. (2003). A kernel-distance-based multivariate control chart using support vector methods. *International Journal of Production Research*, 41(13):2975–2989.
- Sung, S., Lee, J., and Lee, I. (2009). *Process identification and PID control*. John Wiley & Sons.
- Swevers, J., Al-Bender, F., Ganseman, C., and Projogo, T. (2000). An integrated friction model structure with improved presliding behavior for accurate friction compensation. *IEEE Transactions on Automatic Control*, 45(4):675–686. URL <http://dx.doi.org/10.1109/9.847103>.
- Tan, H., Aziz, A. A., and Aroua, M. (2013). Glycerol production and its applications as a raw material: A review. *Renewable and Sustainable Energy Reviews*, 27:118 – 127.
- Tan, K. (1999). *Advances in PID Control*. Advances in Industrial Control. Springer.
- Tang, L., Fang, L., Wang, J., and Shang, Q. (2015). A new semi-physical model for pneumatic control valves with stiction nonlinearity.
- Tangirala, A., Shah, S., and Thornhill, N. (2005). PSCMAP: A new tool for plant-wide oscillation detection. *Journal of Process Control*, 15(8):931 – 941.
- Tangirala, A. K., Kanodia, J., and Shah, S. L. (2007). Non-negative matrix factorization for detection of plant-wide oscillations. *Industrial & Engineering Chemistry Research*, 46(3):801–817.

- Tanner, M. A. and Wong, W. H. (1987). The calculation of posterior distributions by data augmentation. *Journal of the American Statistical Association*, 82(398):528–540.
- Thornhill, N. (2005). Finding the source of nonlinearity in a process with plant-wide oscillation. *Control Systems Technology, IEEE Transactions on*, 13(3):434 – 443.
- Thornhill, N. and Hagglund, T. (1997). Detection and diagnosis of oscillation in control loops. *Control Engineering Practice*, 5(10):1343 – 1354.
- Thornhill, N., Huang, B., and Shah, S. (2003a). Controller performance assessment in set point tracking and regulatory control. *International Journal of Adaptive Control and Signal Processing*, 17(7-9):709–727.
- Thornhill, N., Huang, B., and Zhang, H. (2003b). Detection of multiple oscillations in control loops. *Journal of Process Control*, 13(1):91 – 100.
- Thornhill, N., Oettinger, M., and Fedenczuk, P. (1999). Refinery-wide control loop performance assessment. *Journal of Process Control*, 9(2):109 – 124.
- Thornhill, N., Shah, S., Huang, B., and Vishnubhotla, A. (2002). Spectral principal component analysis of dynamic process data. *Control Engineering Practice*, 10(8):833 – 846.
- Thornhill, N. F., Choudhury, M. S., and Shah, S. L. (2004). The impact of compression on data-driven process analyses. *Journal of Process Control*, 14(4):389 – 398.
- Thornhill, N. F. and Horch, A. (2007). Advances and new directions in plant-wide disturbance detection and diagnosis. *Control Engineering Practice*, 15(10):1196 – 1206. International Symposium on Advanced Control of Chemical Processes ADCHEM.
- Tietjen, G. L. and Moore, R. H. (1972). Some grubbs-type statistics for the detection of several outliers. *Technometrics*, 14(3):583–597.
- Tishler, A. and Zang, I. (1979). A switching regression method using inequality conditions. *J. of Econometrics*, 11(2-3):259–274.
- Trnka, P. (2005). Subspace identification methods. Technical report, Czech Technical University, Prague.
- Tsay, T.-S. (2012). Stability analyses of nonlinear multivariable feedback control systems. *Proceedings of the WSEAS Transactions on Systems*, 11:140–151. URL [www.wseas.org/multimedia/journals/systems/2012/55-169.pdf](http://www.wseas.org/multimedia/journals/systems/2012/55-169.pdf).

## Bibliography

---

- Tsymbal, A. (2004). The problem of concept drift: Definitions and related work. Technical report, Department of Computer Science, Trinity College, Dublin. Consulted in February 2015.
- 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.
- Ulaganathan, N. and Rengaswamy, R. (2008). Blind identification of stiction in nonlinear process control loops. In *American Control Conference, 2008*, pages 3380 –3384.
- Valle, S., Li, W., and Qin, S. J. (1999). Selection of the number of principal components: The variance of the reconstruction error criterion with a comparison to other methods. *Industrial & Engineering Chemistry Research*, 38(11):4389–4401.
- Van den Hof, P. (1998). Closed-loop issues in system identification. *Annual reviews in control*, 22:173–186.
- van Overschee, P. and de Moor, B. (1996). *Subspace Identification for Linear Systems: Theory - Implementation - Applications*. Springer US.
- Vedagarbha, P., Dawson, D., and Feemster, M. (1997). Tracking control of mechanical systems in the presence of nonlinear dynamic friction effects. *Proceedings of the American Control Conference*, 4:2284–2288. URL <http://dx.doi.org/10.1109/ACC.1997.609019>.
- Venceslau, A. R. S. (2012). Detecção e diagnóstico de agarramento em válvulas posicionadoras. Master's thesis, Federal University of Rio Grande do Norte, Rio Grande do Norte (Brazil).
- Venkataramanan, F., Shukla, V., Saini, R., and Rhinehart, R. (1997). An automated online monitor of control system performance. In *American Control Conference, 1997. Proceedings of the 1997*, volume 3, pages 1355 –1359 vol.3.
- Verhaegen, M. and Verdult, V. (2012). *Filtering and System Identification: A Least Squares Approach*. Cambridge University Press.

- Veronesi, M. and Visioli, A. (2010). Performance assessment and retuning of pid controllers for integral processes. *Journal of Process Control*, 20(3):261 – 269.
- Veronesi, M. and Visioli, A. (2015). Deterministic performance assessment and retuning of industrial controllers based on routine operating data: Applications. *Processes*, 3(1):113–137.
- Vijayakumar, S., D’souza, A., and Schaal, S. (2005). Incremental online learning in high dimensions. *Neural Comput.*, 17(12):2602–2634.
- Vilalta, R., Giraud-Carrier, C., and Brazdil, P. (2010). Meta-learning - concepts and techniques. In Maimon, O. and Rokach, L., editors, *Data Mining and Knowledge Discovery Handbook*, pages 717–731. Springer US.
- Villez, K., Srinivasan, B., Rengaswamy, R., Narasimhan, S., and Venkatasubramanian, V. (2010). Resilient control in view of valve stiction: extension of a Kalman-based FTC scheme. volume 28, pages 547–552. Elsevier. URL <http://www.sciencedirect.com/science/article/pii/S1570794610280926>.
- Vörös, J. (2001). Parameter identification of Wiener systems with discontinuous nonlinearities. *Systems & Control Letters*, 44(5):363–372. URL <http://www.sciencedirect.com/science/article/pii/S0167691101001554>.
- Walczak, B. and Massart, D. (2001). Dealing with missing data: Part {II}. *Chemometrics and Intelligent Laboratory Systems*, 58(1):29 – 42.
- Walrath, C. D. (1984). Adaptive bearing friction compensation based on recent knowledge of dynamic friction. *Automatica*, 20(6):717–727. URL <http://www.sciencedirect.com/science/article/pii/0005109884900815>.
- Walther, A. and Griewank, A. (2012). Getting started with ADOL-C. In Naumann, U. and Schenk, O., editors, *Combinatorial Scientific Computing*, chapter 7, pages 181–202. Chapman-Hall CRC Computational Science. URL [http://www2.math.uni-paderborn.de/fileadmin/Mathematik/People/walther/adolc\\_tut.pdf](http://www2.math.uni-paderborn.de/fileadmin/Mathematik/People/walther/adolc_tut.pdf).
- Wang, G. and Wang, J. (2009). Quantification of valve stiction for control loop performance assessment. In *Industrial Engineering and Engineering Management, 2009. IE EM '09. 16th International Conference on*, pages 1189 –1194.

## Bibliography

---

- Wang, J. (2013). Closed-loop compensation method for oscillations caused by control valve stiction. *Industrial & Engineering Chemistry Research*, 0(0):0. URL <http://pubs.acs.org/doi/abs/10.1021/ie400308z>.
- Wang, J., Huang, B., and Lu, S. (2013). Improved DCT-based method for online detection of oscillations in univariate time series. *Control Engineering Practice*, 21(5):622–630. DOI 10.1016/j.conengprac.2012.12.007.
- Wang, J., Sano, A., Chen, T., and Juang, B. (2010). A blind approach to identification of Hammerstein systems. In *IFAC World Congress on Block-oriented Nonlinear System Identification*. Springer.
- Wang, J.-s., Han, S., and Shen, N.-n. (2014). Improved gso optimized esn soft-sensor model of flotation process based on multisource heterogeneous information fusion. *The Scientific World Journal*, 2014(2014).
- Wang, S. and Xiao, F. (2004). AHU sensor fault diagnosis using principal component analysis method. *Energy and Buildings*, 36(2):147 – 160.
- Wang, X., Kruger, U., and Irwin, G. W. (2005). Process monitoring approach using fast moving window PCA. *Industrial & Engineering Chemistry Research*, 44(15):5691–5702.
- Warne, K., Prasad, G., Rezvani, S., and Maguire, L. (2004). Statistical and computational intelligence techniques for inferential model development: a comparative evaluation and a novel proposition for fusion. *Engineering Applications of Artificial Intelligence*, 17(8):871 – 885.
- Watson, M. J., Liakopoulos, A., Brzakovic, D., and Georgakis, C. (1998). A practical assessment of process data compression techniques. *Industrial & Engineering Chemistry Research*, 37(1):267–274.
- Welander, P. (2010). Understanding derivative in PID control. *Control Engineering*.
- Welch, G. and Bishop, G. (2001). *An Introduction to the Kalman Filter*. University of North California, USA.
- Wold, H. (1982). Soft modeling: the basic design and some extensions. In Jöreskog, K. G. and Wold, H., editors, *Proceedings of the Conference on Systems Under Indirect Observation*, volume I and II, Amsterdam (North-Holland).

- Wold, S., Sjöström, M., and Eriksson, L. (2001). PLS-regression: a basic tool of chemometrics. *Chemometrics and Intelligent Laboratory Systems*, 58(2):109 – 130. PLS Methods.
- Wu, Z., Bai †, F., Yang, X., and Zhang, L. (2004). An exact lower order penalty function and its smoothing in nonlinear programming. *Optimization*, 53(1):51–68.
- Wu, Z. Y., Lee, H. W. J., Bai, F. S., and Zhang, L. S. (2005). Quadratic smoothing approximation to  $l_1$  exact penalty function in global optimization. *J. Ind. Manag. Optim.*, 1(4):533–547.
- Xia, C. and Howell, J. (2003). Loop status monitoring and fault localisation. *Journal of Process Control*, 13(7):679 – 691. Selected Papers from the sixth IFAC Symposium on Bridging Engineering with Science - DYCOPS - 6.
- 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 (Safeprocess 2003).
- Xia, C., Howell, J., and Thornhill, N. F. (2005). Detecting and isolating multiple plant-wide oscillations via spectral independent component analysis. *Automatica*, 41(12):2067 – 2075.
- Xia, C., Zheng, J., and Howell, J. (2007). Isolation of whole-plant multiple oscillations via non-negative spectral decomposition. *Chinese Journal of Chemical Engineering*, 15:353–360.
- 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.
- Xu, B., Su, Q., Zhang, J., and Lu, Z. (2015). A dead-band model and its online detection for the pilot stage of a two-stage directional flow control valve. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*.
- Xu, F. and Huang, B. (2006). Performance monitoring of siso control loops subject to ltv disturbance dynamics: An improved lti benchmark. *Journal of Process Control*, 16(6):567 – 579.
- Xu, F., Huang, B., and Tamayo, E. C. (2008). Performance assessment of mimo control systems with time-variant disturbance dynamics. *Computers & Chemical Engineering*, 32(9):2144 – 2154. Networked and Complex Systems S.I. – Control of Networked and Complex Process Systems.

## Bibliography

---

- Xu, S. and Bao, J. (2010). Networked plantwide process control with asynchronous communication and control. In *Proceedings of the 9th International Symposium on Dynamics and Control of Process Systems*, volume 9 of *DYCOPS '10*, pages 85 – 90.
- Yamamoto, S. and Hashimoto, I. (1991). Present status and future needs: the view from Japanese industry. In *Proceedings of the 4th International Conference on Chemical Process Control*, volume TX of *CPC '91*, pages 1–28, Texas (USA).
- Yamashita, Y. (2004). *Qualitative Analysis for Detection of Stiction in Control Valves*, pages 391–397. KES '04. Springer-Verlag GmbH, Wellington (New Zealand), 8th edition edition.
- Yamashita, Y. (2006a). An automatic method for detection of valve stiction in process control loops. *Control Engineering Practice*, 14(5):503 – 510. Intelligent Control Systems and Signal Processing, ICONS. URL <http://www.sciencedirect.com/science/article/pii/S0967066105000833>.
- Yamashita, Y. (2006b). Diagnosis of oscillations in process control loops. In *Proceedings of the European Symposium on Computer Aided Process Engineering*, volume 21 of *Computer Aided Chemical Engineering*, pages 1605–1610. Elsevier. DOI 10.1016/S1570-7946(06)80277-4.
- Yamashita, Y. (2008). Diagnosis and quantification of control valves. *Proceedings of the SICE International Conference on Instrumentation , Control and Information Technology*, pages 2108–2111. URL <http://dx.doi.org/10.1109/SICE.2008.4655010>.
- Yang, H. and Huang, M. (2010). A soft sensor based on kernel pca and composite kernel support vector regression for a flotation circuit. In *Advanced Computer Control (ICACC), 2010 2nd International Conference on*, volume 5, pages 375–378.
- Yang, L. and Sheu, S.-H. (2006). Integrating multivariate engineering process control and multivariate statistical process control. *The International Journal of Advanced Manufacturing Technology*, 29:129–136. 10.1007/s00170-004-2494-8.
- Ye, N. (2003). *The Handbook of Data Mining*. Human factors and ergonomics. Taylor & Francis.
- Yegnanarayana, B. (2004). *Artificial Neural Networks*. Prentice-Hall Of India Pvt. Limited.



- Yu, P., Lu, J., Wu, Y., and Sun, Y. (2006). Soft sensor based on kernel principal component analysis and radial basis function neural network for microbiological fermentation. In *Intelligent Control and Automation, 2006. WCICA 2006. The Sixth World Congress on*, volume 1, pages 4851–4855.
- Zabiri, H., Maulud, A., and Omar, N. (2009). NN-based algorithm for control valve stiction quantification. *Proceedings of the WSEAS Transactions on Systems and Control*, pages 88–97. URL <http://www.wseas.us/e-library/transactions/control/2009/28-918.pdf>.
- Zabiri, H. and Mazuki, N. (2010). A black-box approach in modeling valve stiction. *J. Eng. Appl. Sci.*, 6(5):277–284.
- Zabiri, H. and Ramasamy, M. (2009). Nlpca as a diagnostic tool for control valve stiction. *Journal of Process Control*, 19(8):1368 – 1376. <ce:title>Special Section on Hybrid Systems: Modeling, Simulation and Optimization</ce:title>.
- Zabiri, H. and Samyudia, Y. (2006). A hybrid formulation and design of model predictive control for systems under actuator saturation and backlash. *Journal of Process Control*, 16(7):693 – 709.
- Zabiri, H. and Samyudia, Y. (2009). Miqp-based mpc in the presence of control valve stiction. *Chemical Product and Process Modeling*, 4(3):85–97. URL <http://dx.doi.org/10.2202/1934-2659.1397>.
- Zadeh, L. A. (1956). On the identification problem. *Circuit Theory, IRE Transactions on*, 3(4):277–281.
- Zakharov, A. and Jämsä-Jounela, S.-L. (2014). Robust oscillation detection index and characterization of oscillating signals for valve stiction detection. *Industrial & Engineering Chemistry Research*, 53(14):5973–5981. URL <http://pubs.acs.org/doi/abs/10.1021/ie402636m>.
- Zamprogna, E., Barolo, M., and Seborg, D. E. (2005). Optimal selection of soft sensor inputs for batch distillation columns using principal component analysis. *Journal of Process Control*, 15(1):39 – 52.
- Zang, I. (1981). Discontinuous optimization by smoothing. *Math. Oper. Res.*, 6(1):140–152.

## Bibliography

---

- Zang, X. and Howell, J. (2003). Discrimination between bad turning and non-linearity induced oscillations through bispectral analysis. In *Proceedings of the SICE annual conference, Fukui (Japan)*.
- Zang, X. and Howell, J. (2007). Isolating the source of whole-plant oscillations through bi-amplitude ratio analysis. *Control Engineering Practice*, 15(1):69 – 76.
- Zareba, S., Lakshminarayanan, S., and Jelali, M. (2014). A new controller tuning method based on the relative damping index. In *Proceedings of the 5th International Symposium on Advanced Control of Industrial Processes (ADCONIP)*, Hiroshima (Japan).
- Zeng, J., Xie, L., Gao, C., and Sha, J. (2011). Soft sensor development using non-gaussian just-in-time modeling. In *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, pages 5868–5873.
- Zhang, H. and Lennox, B. (2004). Integrated condition monitoring and control of fed-batch fermentation processes. *Journal of Process Control*, 14(1):41 – 50.
- Zhang, Y. and Henson, M. A. (1999). A performance measure for constrained model predictive controllers. In *Proceedings of the European Control Conference, Karlsruhe (Germany)*.
- Zhang, Y. and Zhang, Y. (2010). Process monitoring, fault diagnosis and quality prediction methods based on the multivariate statistical techniques. *IETE Technical Review*, 27(5):406 – 420.
- Zhang, Z. (1997). Parameter estimation techniques: a tutorial with application to conic fitting. *Image and Vision Computing*, 15(1):59 – 76. URL <http://www.sciencedirect.com/science/article/pii/S0262885696011122>.
- ZHANG Wenqing, FU Yujia, Y. H. (2012). Multi-model soft-sensor modeling based on improved clustering and weighted bagging. *CIESC Journal*, 63(9):2697.
- Zheng, Q. and Kimura, H. (2001). Just-in-time modeling for function prediction and its applications. *Asian Journal of Control*, 3(1):35–44.
- Zhu, Y. (2001). *Multivariable System Identification For Process Control*. Pergamon.
- Ziegler, J. and Nichols, N. (1942). Optimum settings for automatic controllers. *Transactions of the ASME Journal*, 64:759 – 768.

# Appendices



# Appendix A

## CSTR model

Consider a simple liquid phase reactor where an irreversible first-order chemical reaction takes place converting reactant A to product B. The inlet stream consists of pure component A with molar concentration  $C_{Ai}$ . A heating coil is used to maintain the reaction mixture at the desired operating temperature by adding heat needed for the endothermic reaction to take place.

A deterministic mathematical model can be built based on the following assumptions:

- the CSTR is perfectly mixed;
- the reaction rate can be defined using the Arrhenius equation:  $k = k_0 \exp(-\frac{E_a}{RT})$ , where  $k_0$  is the frequency factor,  $E_a$  is the activation energy and  $R$  is the gas constant;
- the mass densities,  $\rho$ , and the specific heat capacity,  $c_p$ , of the feed and product streams are equal and constant;
- the liquid volume,  $V$ , in the reactor is kept constant;
- the thermal capacitances of the heating fluid and of the coil wall are negligible compared to the thermal capacitance of the liquid in the tank;
- all the heating fluid is at a uniform temperature,  $T_c$ ;
- the rate of heat transfer from the heating fluid to the reacting mixture is given by  $UA_t(T_c - T)$  where  $U$  is the overall heat transfer coefficient and  $A_t$  is the heat transfer area.

Table A.1: CSTR model parameters.

Parameter	Value	Unit
$A_t$	9.7980	$\text{m}^2$
$c_p$	1033.78	$\text{J kg}^{-1} \text{ }^\circ\text{C}^{-1}$
$E_a/R$	$1.0838 \cdot 10^4$	$^\circ\text{C}$
$k_0$	$4.0 \cdot 10^{13}$	$\text{s}^{-1}$
$q$	0.0013	$\text{m}^3 \text{ s}^{-1}$
$T_i$	50	$^\circ\text{C}$
$U$	500	$\text{W m}^{-2} \text{ }^\circ\text{C}^{-1}$
$V$	3.7854	$\text{m}^3$
$\Delta H$	$5.0 \cdot 10^5$	$\text{J mol}^{-1}$
$\rho$	832.96	$\text{kg m}^{-3}$

Material and energy balance equations may be rearranged into

$$V \frac{dC_A}{dt} = q(C_{Ai} - C_A) - V k C_A, \quad (\text{A.1a})$$

$$V \rho c_p \frac{dT}{dt} = q \rho c_p (T_i - T) + (-\Delta H) V k C_A + U A_t (T_c - T). \quad (\text{A.1b})$$

The model parameters were adapted from exercise 4.14 of Seborg et al. (2010) and are listed in Table A.1.

The set of differential equations defined by (A.1) was implemented in GNU Octave (Eaton, 2002) and integrated using LSODE solver (Hindmarsh, 1983) for a series of different steps in the input variables profiles and with a finite-differences approximation of the derivative information. More details about it can be found in Radhakrishnan and Hindmarsh (1993). The outputs of the model, corrupted with a random noise, constitute the experimental dataset for the identification of the MIMO system.

# Appendix B

## Benchmarks modifications and extensions

This appendix summarizes the modifications and extensions that have taken place in some benchmarks and indices. Modifications and extensions of the MVC benchmark are introduced in Table B.1, of the user-specified indices in Table B.2, and of the model-based benchmarks in Table B.3.

Table B.1: Modifications and extensions of the MVC benchmark.

Work	Modification/Extension
<i>SISO performance index</i>	
Desborough and Harris (1992)	connected the Harris index to the squared correlation coefficient usually calculated by multiple regression analysis
Tyler and Morari (1995) and Tyler and Morari (1996)	extended to unstable and non-minimum-phase systems and introduced statistical likelihood ratio tests
Lynch and Dumont (1996)	used Laguerre networks to evaluate the performance index
Huang and Shah (1998)	developed an efficient, stable filtering and correlation method to estimate MVC
Qin (1998)	derived the minimum variance control based performance assessment using the internal model control structure
Desborough and Harris (1993), Stanfelj et al. (1993) and Huang et al. (2000a)	extended to feedback and feedforward control loops
McNabb and Qin (2003)	built a state-space framework for MVC

Continued on next page

**Table B.1 – continued from previous page**

Work	Modification/Extension
<i>MIMO performance index</i>	
Huang and Shah (1998)	showed that the performance assessment of MIMO feedback systems can be estimated from closed-loop data
Harris et al. (1996), Huang and Shah (1998) and Huang and Shah (1999)	showed the important role of interaction matrix in the performance assessment of MIMO feedback systems
Ettaleb (1999)	suggested a practical solution to MIMO control performance assessment (for minimum phase systems) that performs, in a first stage, time-series analysis as in SISO case independently for each output $y_i$ to get minimum achievable output variance $\sigma_{MV,i}^2$ .
Ettaleb (1999), Ko and Edgar (2001b) and McNabb and Qin (2003)	studied the elimination of the requirement to develop the interaction matrix for multivariate systems with characteristics that lead to a diagonal interaction matrix
Huang et al. (2005)	investigated a multivariable control benchmark which needs only the order of the interaction matrix. This approach is simple and a natural extension of SISO control performance assessment resulting in a less aggressive benchmark control than the later approaches.
<i>Varying the setpoint of the controlled variables</i>	
Perrier and Roche (1992) and Ko and Edgar (2000)	modified the minimum variance index to include setpoint variations in the inner loop of cascade controls
Seppala et al. (2002)	discussed the influence of setpoint changes on MVC and demonstrated the benefits of a decomposition of the control error into the components resulting from setpoint changes and setpoint detrended signal
Thornhill et al. (2003a)	examined the reasons why performance during setpoint changes differs from regulatory performance during operation at a constant setpoint
<i>Processes with time-variant behavior</i>	
Huang (1999, 2002)	built a general framework for control performance assessment of linear time-variant processes
Gustafsson (2000), Salsbury (2005) and Xia et al. (2005)	described methods for detecting disturbance/load changes and characterizing subsequent responses in single-input-single-output (SISO).

Continued on next page



---

**Table B.1 – continued from previous page**

<b>Work</b>	<b>Modification/Extension</b>
Xu and Huang (2006)	considered performance analysis problems for the process that is subject to time varying disturbance dynamics formulating the problem as the minimization of the sum of the weighted variances of all disturbances.
Xu et al. (2008)	extended Xu and Huang (2006) approach to MIMO systems.

**Table B.2: Modifications and extensions of the user-specified performance indices.**

<b>Work</b>	<b>Observations</b>
Desborough and Harris (1992) and Thornhill et al. (1999)	proposed the use of the extended horizon performance index to consider user specifications and/or to avoid requiring the loop time delay for calculating the control performance index.
Rhinehart (1995) and Venkataraman et al. (1997)	proposed and applied the statistical test r-statistic which detects deviations from a setpoint regardless of the output-noise amplitude.
Bezergianni and Georgakis (2000, 2003)	introduced the relative variance performance index that compares actual control to both minimum variance control and open-loop control.
Li et al. (2003)	presented the relative performance monitor index which compares the performance of a control loop to that of a reference model.
Hagglund (2005)	suggested to monitor the extended horizon performance index and an alert limit according to the tuning of the loop in order to fulfill design specifications.

Table B.3: Modifications and extensions of the model-based performance indices.

Work	Observations
Eriksson and Isaksson (1994) and Ko and Edgar (1998)	applied PID controller order, structure and action constraints for the calculation of the more realistic performance indicator optimal PID benchmarking.
Zhang and Henson (1999)	proposed expectation-case approach where the actual performance is compared online to the expected performance obtained when the controller actions are implemented on the process model instead of the plant.
Ko and Edgar (2001a)	developed a methodology for the estimation of a constrained minimum variance performance bound for MIMO systems with stable process inverses. The performance bounds were subsequently used for the performance assessment of MPC.
Horton et al. (2003) and Huang (2003)	developed the approach of Ko and Edgar (2004) requiring the process/disturbance model knowledge/identification and the use of optimization algorithms to calculate the optimal controller settings.
Grimble (2003)	provided a theoretical framework for control performance analysis based on state-space models.
Ko and Edgar (2004)	derived an explicit "one-shot" solution for the closed-loop output as function of PID parameters.
Ko and Edgar (2001a) and Schafer and Cinar (2004)	developed a model-based approach for benchmarking of MPC systems, which can explicitly handle long-time delays and constraints.
Julien et al. (2004)	proposed design-case benchmark showing to be useful both as a model diagnostic and as a tuning guide during commissioning.
Rato (2009)	normalizes the index developed by Qin and Yu (2007) to contemplate variations in the load disturbances enabling less variation and lower false alarm rate while maintains its ability to detect the controller performance deterioration.

# Appendix C

## Missing data

This appendix summarizes some techniques to assess missingness mechanisms. It also provides two techniques to handle missing data based on the maximum likelihood method and on the multiple imputation method. It also exemplifies the application of several techniques using a small dataset.

### C.1 Methods for assessing MCAR mechanisms

The *univariate t-test based approach* (Dixon et al., 1988) separates the missing and the complete cases of a particular variable  $X_1$  into two subgroups. If MCAR mechanism is present, both subgroup means are the same. A *t*-test is applied to examine subgroup mean differences on the other variable of the dataset,  $X_2$ . The algorithm is described as follows:

1. Select variables  $X_1$  and  $X_2$ .
2. Calculate  $M$  matrix.
3. Using just the complete samples of  $X_2$ , separate the missing and complete cases of  $X_1$  into two subgroups:  
 $X_{\text{obs}} = \{X_{1,\text{obs}}, X_{2,\text{obs}}\}$  and  $X_{\text{miss}} = \{X_{1,\text{miss}}, X_{2,\text{obs}}\}$ .
4. Calculate the mean of  $X_2$  for each subgroup.
5. Perform the *t*-test to examine subgroup mean differences.
6. If  $p < 0.01$ , the mean difference is statistically significant and variable  $X_1$  is considered not MCAR.
7. If  $p > 0.01$ , the mean difference is not statistically significant and variable  $X_1$  is considered MCAR.

The correlation among variables may influence the test results because significant mean differences may be detected in several variables when the real cause arises from just one of the variables (Enders, 2010).

The *multivariate Little MCAR test* (Little, 1988) compares the subgroups means to the maximum likelihood estimates of the grand means. Contrarily to the previous test, it uses the cases that have a same missing data pattern applying the statistic test

$$d^2 = \sum_{j=1}^J d_j^2 = \sum_{j=1}^J n_j \left( \hat{\mu}_j - \hat{\mu}_j^{\text{ML}} \right)^\top \hat{\Sigma}_j^{-1} \left( \hat{\mu}_j - \hat{\mu}_j^{\text{ML}} \right), \quad (\text{C.1})$$

where  $J$  is the total number of missing patterns,  $n_j$  is the number of the cases in the missing pattern  $j$ ,  $\hat{\mu}_j$  is the mean vector for the cases in the missing pattern  $j$ ,  $\hat{\mu}_j^{\text{ML}}$  is the grand mean vector estimated by the maximum likelihood method, and  $\hat{\Sigma}_j$  is the covariance matrix estimated by the maximum likelihood method.  $d^2$  statistic is essentially a weighted sum of the  $J$  squared differences between pattern  $j$  means and the corresponding grand means. When the data presents a MCAR missing mechanism, this statistic has approximately a  $\chi^2$ -distribution with  $\sum_{j=1}^J k_j - k$  degrees of freedom.

The multivariate Little MCAR test algorithm is:

1. Estimate the maximum likelihood parameters  $\hat{\mu}_j^{\text{ML}}$  and  $\hat{\Sigma}_j$ .
2. Select a missing pattern.
3. Calculate the statistic test  $d_j^2$  for the selected missing pattern.
4. Repeat steps 1 and 2 for every missing patterns.
5. Perform the  $\chi^2$ -test to the value  $d^2$  with  $\sum_{j=1}^J k_j - k$  degrees of freedom, where  $k_j$  is the number of complete variable for missing pattern  $j$ , and  $k$  is the total number of variables.
6. If  $p < 0.01$ , the mean differences are statistically significant and the data are considered not MCAR.
7. If  $p > 0.01$ , the mean differences are not statistically significant and the data are considered MCAR.

## C.2 Methods for handling missing data

The *maximum likelihood method*, often referred to as full information maximum likelihood or direct maximum likelihood method, is a missing data handling method whose formulation relies on a probability density function to describe the population data. Researchers routinely assume that the variables are normally distributed in the population and the density function comes as

$$L_i = \frac{1}{(2\pi)^{k_i/2} |\boldsymbol{\Sigma}_i|^{1/2}} \exp \left[ -0.5 (\mathbf{Y}_i - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}_i^{-1} (\mathbf{Y}_i - \boldsymbol{\mu}) \right], \quad (\text{C.2})$$

where  $\mathbf{Y}_i$  are the sample scores,  $\boldsymbol{\mu}$  is the population mean vector,  $\boldsymbol{\Sigma}$  is the population covariance matrix,  $L_i$  is the likelihood value that describes the height of the normal curve at a particular score value, and  $k_i$  is the number of complete data points for the case.

In order to identify the population parameter values that have the highest probability of producing a particular sample of data, the maximum likelihood is estimated. The fit measure used to estimate these values is the log-likelihood consisting of the sum of the individual log-likelihood values, that is,

$$\log L = \sum_{i=1}^N \log L_i, \quad (\text{C.3})$$

with  $L_i$  given by C.2.

The algorithm of the maximum likelihood method is:

1. Select a missing pattern.
2. Calculate the individual log-likelihood of the cases contained in the selected missing pattern considering just the population parameter values for the observed data in that pattern.
3. Iterate along the missing patterns.
4. Calculate  $\log L$  summing all the individual log-likelihoods.
5. Readjust the population parameter values and run again steps 1 to 4 in order to maximize the log-likelihood  $\log L$ .

In the context of the maximum-likelihood method, other algorithms can be used such as the expectation-maximization algorithm proposed by Dempster et al. (1977) which is a deterministic iterative approach. The algorithm is composed by the expectation and maximization steps (E-step and M-step, respectively). The iterative algorithm starts with estimates of the mean vector and covariance matrix (considering, for instance, the list-wise deletion method). The E-step builds a set of regression equations that predict the incomplete variables using these estimates, while the M-step generates updated estimates for the mean vector and the covariance matrix in such a way that the  $\log L$  is maximized. These steps are repeated until the convergence of the mean vector and covariance matrix are reached (Ghomrawi et al., 2011). It is worth mentioning that the expectation-maximization algorithm may diverge and find solutions with infinite likelihood unless the covariances are regularized artificially.

The *multiple imputation method* is a modern tool for handling missing data which generates several copies of the dataset and fills the gaps of missing values in each copy with different estimates. The method consists of the imputation, the analysis, and the pooling phases. The imputation phase generates  $m$  copies of the dataset containing different estimates of the missing values, the analysis phase applies the soft sensor development method to each generated copy, and the pooling phase combines the different results into a single final result.

Several methods based on the Markov chain Monte Carlo are used to implement the multiple imputation method, namely Gibbs sampling, data augmentation, the Metropolis-Hastings algorithm, among others (Schafer, 1997, Section 1.2). One of the most popular is the data augmentation algorithm developed by Tanner and Wong (1987). It is an iterative method that repeatedly cycles between two steps: the imputation step (I-step) and the posterior step (P-step). Basically, the I-step uses the stochastic regression imputation method to fill in the missing values, while the P-step characterizes the posterior distribution and uses a Monte Carlo simulation to draw new estimates for the defined distribution. The P-step consists in a standalone Bayesian analysis (Enders, 2010). Several copies of the data are obtained when this two-step procedure is run repeatedly.

Suppose that the complete data  $Y$  may be written as  $Y = \{Y_{\text{obs}}, Y_{\text{miss}}\}$  and that

the missing mechanism is ignorable. Given an estimation of  $\theta_t^*$ , the missing data  $Y_{\text{miss}}^*$  is imputed from the distribution

$$Y_{\text{miss}}^* \sim p(Y_{\text{miss}} | Y_{\text{obs}}, \theta_t^*). \quad (\text{C.4})$$

This step is the so-called I-step. Considering now the complete dataset  $Y^* = \{Y_{\text{obs}}, Y_{\text{miss}}^*\}$ , the estimation of  $\theta_{t+1}^*$  is generated from

$$\theta_{t+1}^* \sim p(\theta_{t+1} | Y_{\text{obs}}, Y_{\text{miss}}^*). \quad (\text{C.5})$$

Procedurally, a Monte Carlo simulation randomly generates new parameters values (also called simulated parameters),  $\theta_{t+1}^* = \{\mu_{t+1}^*, \Sigma_{t+1}^*\}$ , using the following posterior distributions

$$\Sigma_{t+1}^* \sim p(\Sigma | \hat{\mu}, Y^*) \sim W^{-1}(N - 1, \hat{\Lambda}), \quad (\text{C.6})$$

and

$$\mu_{t+1}^* \sim p(\mu | Y^*, \Sigma_{t+1}^*) \sim MN(\hat{\mu}, N^{-1}\Sigma_{t+1}^*), \quad (\text{C.7})$$

where  $\hat{\mu}$  is the sample mean vector,  $\hat{\Sigma}$  is the sample covariance matrix,  $W^{-1}$  is the inverse Wishart distribution,  $N$  is the number of samples,  $\hat{\Lambda}$  is the sample sum of squares and cross products matrix,  $\hat{\Lambda} = (N - 1)\hat{\Sigma}$ ,  $MN$  denotes a multivariate normal distribution,  $\mu_{t+1}^*$  is the simulated mean vector, and  $\Sigma_{t+1}^*$  is the simulated covariance matrix. This step is called P-step.

Both I- and P-steps are performed iteratively until a stopping criterion has been satisfied. The imputed dataset must mimic independent draws from the distribution of the missing values. These data can be generated either by the sequential or the parallel data augmentation chains. While the sequential strategy selects a dataset at regular intervals in the data augmentation chain (for example, it selects the imputed dataset from every 150 iterations), the parallel strategy generates several data augmentation chains and chooses the final imputed dataset of each chain (Enders, 2010, Section 7.13). The total number of iterations needed in a given application to generate independent draws is problematic because it often requires huge computation time (Kim and Shao, 2013).

Data augmentation generates random parameter estimates across successive

P-steps that have been very hard to converge to a stable posterior distribution. This random behaviour adds a layer of complexity that was not present in approaches such as the maximum likelihood estimation (Enders, 2010). There are several methods for assessing the convergence of data augmentation. The use of graphical displays, specially the time-series and autocorrelation function plots, are methods commonly used due to their ready availability in software packages.

### C.3 Application to a small dataset

Missing data handling is illustrated using variables  $A$ ,  $B$  and  $C$  of the small dataset in Table C.1. Scores of variable  $B$  were randomly deleted in order to mimic a MCAR mechanism. In addition and to simulate a MAR mechanism, the scores of variable  $C$  were systematically missing for variable  $A$  scores contained in the lower half of  $A$  distribution.

#### Assessing the MCAR mechanism

The univariate  $t$ -test based approach is applied to assess the presence of MCAR mechanism in variables  $A$ ,  $B$  and  $C$ . The following combinations of data

$$(X_1, X_2) = \{(C, A), (C, B), (B, A), (B, C)\}$$

are evaluated. Table C.2 presents the results of the test application to the dataset.

Focusing on variable  $C$ , the  $t$ -test applied to the pair  $(C, A)$  indicates that the means of the subgroups are statistically different ( $p < 0.001$ , Table C.2), suggesting that variable  $C$  is not MCAR. This conclusion is correct because the missing values of the variable  $C$  were selected based on the distribution of variable  $A$ . Contrarily, the  $t$ -test for the pair  $(C, B)$  indicates that the mean difference is not significant ( $p = 0.19$ ) which supports the claim that the missingness mechanism of  $C$  is MCAR. Collectively, these tests suggest that variable  $C$  is not MCAR because there is a dependence of the missing values on other variables (variable  $A$ ).

Evaluating now variable  $B$ , the  $t$ -test to  $(B, A)$  indicates that the subgroup means are equivalent, providing support to the MCAR mechanism. Finally, the comparison between variables  $B$  and  $C$  is not performed because there is only



Table C.1: Dataset (Enders, 2010, Table 1.1).

$A$ (complete)	$B$ (MCAR mech.)	$C$ (MAR mech.)	$C$ (complete)
78	13	–	9
84	9	–	13
84	10	–	10
85	10	–	8
87	–	–	7
91	3	–	7
92	12	–	9
94	3	–	11
94	13	–	11
96	–	–	7
99	6	7	7
105	12	10	10
105	14	11	11
106	10	15	15
108	0	10	10
112	10	10	10
113	14	12	12
115	14	14	14
118	12	16	16
134	11	12	12

one case of the missing data subgroup. Consequently, the missingness mechanism of variable  $B$  is MCAR which is in agreement with the way used to choose the missing values.

To illustrate the multivariate Little MCAR test, reconsider the dataset in Table C.1 that contains three variables ( $k = 3$ ) and, therefore, may present four missing patterns ( $J = 4$ ): cases with only  $A$  scores, cases with  $A$  and  $B$  scores, cases with  $A$ ,  $B$  and  $C$  scores, and cases with  $A$  and  $C$  scores. Matrices generated by the maximum likelihood method are

$$\hat{\boldsymbol{\mu}}^{\text{ML}} = \begin{bmatrix} 100.00 \\ 10.23 \\ 10.27 \end{bmatrix} \text{ and } \hat{\boldsymbol{\Sigma}} = \begin{bmatrix} 189.60 & 12.21 & 22.31 \\ 12.21 & 11.04 & 6.50 \\ 22.31 & 5.61 & 8.68 \end{bmatrix}.$$

Table C.3 shows the considered patterns, the corresponding statistic variables  $d_j^2$ ,

Table C.2: Results of the univariate  $t$ -test based approach.

$(X_1, X_2)$	$\bar{X}_2$ in $X_{\text{obs}}$	$\bar{X}_2$ in $X_{\text{miss}}$	degrees of freedom	$t$	$p$
$(C,A)$	111.50	88.50	14.68	6.44	$10^{-5}$
$(C,B)$	11.44	9.13	11.70	1.39	0.19
$(B,A)$	100.53	97.00	3.60	0.50	0.65
$(B,C)$	–	–	–	–	–

Table C.3: Results of the multivariate Little MCAR test.

$j$	Missing pattern	$d_j^2$	$n_j$	$k_j$
1	$\{A_{\text{obs}}, B_{\text{miss}}, C_{\text{miss}}\}$	0.762	2	1
2	$\{A_{\text{obs}}, B_{\text{obs}}, C_{\text{miss}}\}$	6.431	8	2
3	$\{A_{\text{obs}}, B_{\text{obs}}, C_{\text{obs}}\}$	6.868	9	3
4	$\{A_{\text{obs}}, B_{\text{miss}}, C_{\text{obs}}\}$	0.564	1	2
		$d^2 =$	14.625	

and other auxiliary variables. Performing the  $\chi^2$ -test to the value  $d^2$  with  $\sum_{j=1}^J k_j - k = 5$  degrees of freedom, the probability  $p = 0.005$  indicates that the mean differences are statistically significant and the data are not MCAR.

Unlike the previous test, the multivariate Little MCAR test did not identify the specific variables that violate MCAR mechanism.

### Applying the maximum likelihood method based on numerical optimization

The maximum likelihood method in missing data is exemplified below using the data in Table C.1. Estimating the statistical parameters mean and covariance is easy with complete data. But, with incomplete data, this task will require an iterative method.

Table C.4 shows the maximum likelihood and list-wise methods estimates and the statistical parameters for the complete data. Because the list-wise deletion method discards half of the samples, the remaining cases are unrepresentative of the data and, consequently, estimates are too different. In contrast, estimates by the maximum likelihood method are relatively similar to those of the complete data. The maximum likelihood method produces better estimates of the

Table C.4: Results of the maximum likelihood method based on numerical optimization and comparison with other methods.

Estimator	$\mu_A$	$\mu_C$	$\sigma_A^2$	$\sigma_C^2$	$\sigma_{AC}$
Complete data	100.00	10.35	189.60	6.83	19.50
Maximum likelihood method	100.00	10.28	189.60	8.21	23.41
List-wise deletion method	111.50	11.70	84.65	6.61	10.45

means because it uses the incomplete cases while list-wise deletion method ignores them.

### Applying the multiple imputation method based on augmentation algorithm

To illustrate the multiple imputation method application using data augmentation, first and third columns of Table C.1 are used (variables  $A$  and  $C$ ). In addition to the complete pattern  $\{A_{\text{obs}}, C_{\text{obs}}\}$ , there is the missing data pattern  $\{A_{\text{obs}}, C_{\text{miss}}\}$  where only  $A$  is observed. When there are multiple missing data patterns, the imputation process complicates because each missing data pattern needs a unique regression equation (Enders, 2010, p. 200, Table 7.5).

Firstly, the I-step constructs the regression equation

$$\hat{B}_i = \hat{\beta}_0 + \hat{\beta}_1 A_i \quad (\text{C.8})$$

with  $\hat{\beta}_0 = -2.06462$  and  $\hat{\beta}_1 = 0.12345$ . Then, the algorithm generates predicted values by substituting the observed data  $A_i$  into the regression equation, and it augments each predicted score  $\hat{B}_i$  with a normally distributed residual term  $z_i$  with zero mean and residual variance equals to 5.9111. Table C.5 shows the  $m = 5$  imputed datasets using as starting point both mean vector and covariance matrix calculated by the list-wise deletion method. Using the series data augmentation chain, each dataset was collected after 150 cycles.

In some situations, data augmentation fails to converge. For example, when some parameters are inestimable or because the number of variables is close to the number of cases. The first solution consists of eliminating the problematic variables but this is not the ideal solution because this may alter the quality of the data. An alternative solution, called ridge prior method, consists of adding

Table C.5: Imputed dataset s using data augmentation..

<i>A</i>	<i>C</i>				
	<i>m</i> = 1	<i>m</i> = 2	<i>m</i> = 3	<i>m</i> = 4	<i>m</i> = 5
78	5.6178	4.9854	10.2433	5.9090	5.9090
84	6.0314	8.4913	9.6137	5.6551	5.6551
84	8.2116	11.3109	7.2158	9.6158	9.6158
85	11.3865	10.2372	7.1209	10.8447	10.8447
87	9.6068	9.8072	6.0376	12.1948	12.1948
91	7.5882	12.6001	12.7899	5.1053	5.1053
92	11.7692	5.9490	8.5538	11.4526	11.4526
94	12.0710	7.6582	5.7982	8.5460	8.5460
94	5.2938	7.0966	12.0719	10.3325	10.3325
96	11.0303	10.4707	9.1615	8.9509	8.9509
99	7	7	7	7	7
105	10	10	10	10	10
105	11	11	11	11	11
106	15	15	15	15	15
108	10	10	10	10	10
112	10	10	10	10	10
113	12	12	12	12	12
115	14	14	14	14	14
118	16	16	16	16	16
134	12	12	12	12	12

a small number of imaginary data records from a hypothetical population where the variables are uncorrelated to stabilize the estimation and eliminate convergence problems. The convergence for the mean vector and covariance matrix may be inspected using a time-series plot. Figure C.1 exhibits the evolution of the mean vector estimates. It is easy to realize that the two simulated parameters settled into a random pattern almost immediately.

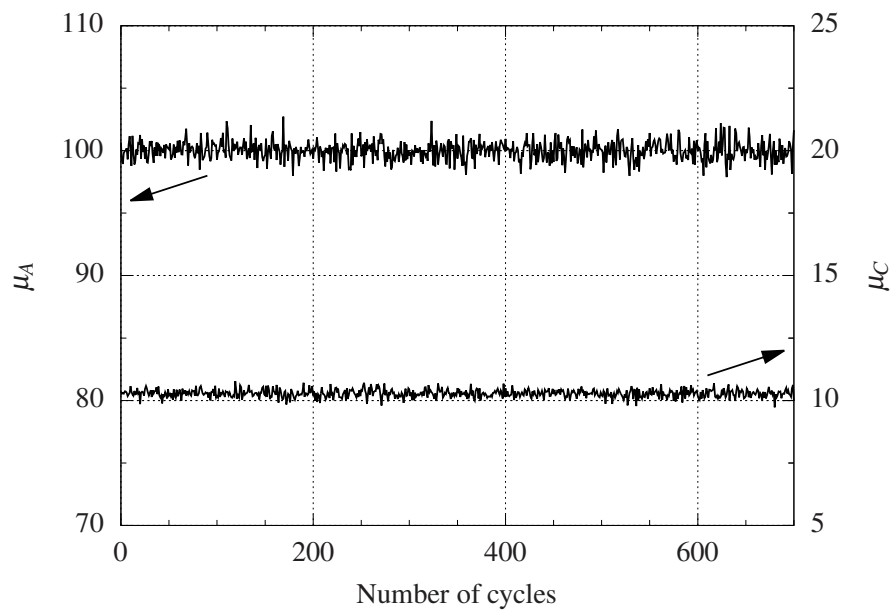


Figure C.1: Time-series plot for the simulated  $A$  and  $C$  means.







UNIVERSITY OF COIMBRA

**Bibtex entry:**

```
@PhdThesis{brasio15,  
  Author = {Brásio, Ana SR},  
  Title = {Industrial processes monitoring methodologies},  
  School = {University of Coimbra},  
  Address = {Coimbra (Portugal)},  
  Year = 2015  
}
```