



Universitat de Girona

# A PROPOSAL FOR THE DIAGNOSIS OF UNCERTAIN DYNAMIC SYSTEMS BASED ON INTERVAL MODELS

**Esteban Reinaldo GELSO**

**ISBN: 978-84-692-4278-0**

**Dipòsit legal: GI-852-2009**

Universitat de Girona  
Departament d'Enginyeria Elèctrica, Electrònica i Automàtica



# A proposal for the diagnosis of uncertain dynamic systems based on interval models

Esteban Reinaldo Gelso

Doctoral Thesis

*Supervisor*

*Dr. Joaquim Armengol*

Girona, Spain  
April 2009



Universitat de Girona  
Departament d'Enginyeria Elèctrica, Electrònica i Automàtica

# **A proposal for the diagnosis of uncertain dynamic systems based on interval models**

A dissertation presented to the Universitat de  
Girona in partial fulfillment of the requirements  
of the degree of DOCTOR OF PHILOSOPHY

By

Esteban Reinaldo Gelso

Supervisor

Dr. Joaquim Armengol

Girona, Spain

April 2009



To my parents, Rosita and Reinaldo,  
to the memory of my grandfather, José,  
to my siblings, Rosana and Sebastián,  
and to Sandra, with love.



## Acknowledgements

I would like to acknowledge the financial support of the Government of Catalonia (Agency for Administration of University and Research Grants - AGAUR), particularly in the award of the following grants: pre-doctorate grant 2005FI00682 from 2005 to 2008, and the BE grants 2006-BE2-00089, 2007-BE2-00311, and 2008-BE1-00186, to carry out research stays in Sweden, at Linköping University, in USA, at Vanderbilt University, and in Denmark, at Technical University of Denmark, respectively. This work was supported also by the Spanish Government through the coordinated research projects No. DPI2003-07146-C02-02 and No. DPI2006-15476-C02-02.

I would like to express my gratitude and thanks to my supervisor, Professor Joaquim Armengol, for the guidance and support he provided me throughout the process of writing this thesis.

I would also like to thank my mentor, Professor Belarmino Pulido, who introduced me the fascinating world of fault diagnosis, and who taught me many valuable things essential to this stage.

I am grateful to Professor Gautam Biswas, Professor Mogens Blanke, and Professor Erik Frisk, for giving me the opportunity to work with them during my research stays.

I would also like to thank the reviewers of the thesis, Professor Louise Travé-Massuyès and Professor Sylviane Gentil.

Also, I would like to thank members of the following research groups I have worked with during my doctoral studies, Engineering Faculty of Olavarría (National University of the Center of Buenos Aires Province, Argentina), GSI Group-Computer Science Department (University



of Valladolid, Spain), Modal Intervals and Control Engineering Research Group (University of Girona, Spain), Division of Vehicular Systems-Department of Electrical Engineering (Linköping University, Sweden), Modeling and Analysis of Complex Systems Group-Institute for Software Integrated Systems (Vanderbilt University, USA), and Department of Electrical Engineering (Technical University of Denmark, Denmark). It is impossible to mention all the persons and I would not like to forget any of them.

All the work put into this thesis would not have been possible without someone at work and at home to lean upon. Thank you, Sandra, for your love and support during these years of research, and for putting everything into perspective. I am deeply grateful to her for being the loving and generous person she is.

Last, but by no means least, I would like to express my gratitude to all my family and friends for their support and patience, and specially to my mother Rosita Bagnoli, my father Reinaldo Gelso, my brother Sebastián Gelso, and my sister Rosana Gelso, for their continued support in all my choices, despite the great distance between us.

Girona, April 2009  
Esteban R. Gelso

## **Abstract**

The performance of a model-based diagnosis system could be affected by several uncertainty sources, such as, model errors, uncertainty in measurements, and disturbances. This uncertainty can be handled by mean of interval models.

The aim of this thesis is to propose a methodology for fault detection, isolation and identification based on interval models. Especially a modal interval based technique, and interval based consistency technique (ICTs) are considered. The thesis includes some proposals to perform each stage within the diagnosis process, from the design of the fault detection tests, to the quantitative fault isolation and identification stage.

Thus, the methodology includes some algorithms to obtain in an automatic way the symbolic expression of the residual generators enhancing the structural isolability of the faults, in order to design the fault detection tests. These algorithms are based on the structural model of the system.

The stages of fault detection, isolation, and identification are stated as constraint satisfaction problems in continuous domains and solved by means of interval based consistency techniques. The qualitative fault isolation is enhanced by a reasoning in which the signs of the symptoms are derived from analytical redundancy relations or bond graph models of the system. The quantitative fault isolation and identification is performed by means of a faulty parameters estimation approach.

An initial and empirical analysis regarding the differences between interval-based and statistical-based techniques is presented in this thesis. Methods included in this comparison are, ICTs on one side, and a statistical decision technique that combines Extended Kalman filter and Z-test on the other side.

The performance and efficiency of the contributions are illustrated through several application examples, covering different levels of complexity.

## Resumen

Las prestaciones de un sistema de diagnóstico basado en modelos se pueden ver afectadas por fuentes de incertidumbre como los errores en el modelo, la incertidumbre en las medidas y las perturbaciones. Esta incertidumbre se puede tratar mediante modelos intervalares.

Esta tesis propone una metodología de detección, aislamiento e identificación de fallos basada en modelos intervalares. En concreto, se consideran técnicas basadas en intervalos modales y técnicas de consistencia basadas en intervalos (*Interval based Consistency Techniques*, ICT). La tesis incluye propuestas en cada etapa del proceso de diagnóstico, desde el diseño de las pruebas de detección de fallos, hasta las etapas de aislamiento cuantitativo e identificación de fallos.

Así, la metodología incluye algoritmos para obtener de manera automática la expresión simbólica de los generadores de residuos mejorando la aislabilidad estructural de los fallos. Estos algoritmos se basan en el modelo estructural del sistema.

Las etapas de detección, aislamiento, e identificación de fallos se representan como problemas de satisfacción de restricciones en dominios continuos y se resuelven por medio de técnicas de consistencia basadas en intervalos. Una mejora en el aislamiento cualitativo de los fallos se obtiene por razonamiento con los signos de los síntomas, que se obtienen de relaciones de redundancia analítica o de modelos *bond graph* del sistema. El aislamiento cuantitativo y la identificación de los fallos se realiza por medio de la estimación de los parámetros que fallan.

Esta tesis también presenta un análisis empírico inicial de las diferencias entre las técnicas basadas en intervalos y las basadas en

técnicas estadísticas. Los métodos incluidos en esta comparación son las técnicas ICT por una parte y por otra el filtro extendido de Kalman y la técnica de decisión estadística *Z-test*.

Las prestaciones y la eficiencia de las contribuciones de la tesis se ilustran a través de unos cuantos ejemplos de aplicación, que cubren diferentes niveles de complejidad.

## Resum

Les prestacions d'un sistema de diagnosi basat en models es poden veure afectades per fonts d'incertesa com els errors en el model, la incertesa en les mesures i les pertorbacions. Aquesta incertesa es pot tractar mitjançant models intervalars.

Aquesta tesi proposa una metodologia de detecció, aïllament i identificació de falles basada en models intervalars. En concret, es consideren tècniques basades en intervals modals i tècniques de consistència basades en intervals (*Interval based Consistency Techniques, ICT*). La tesi inclou propostes en cada etapa del procés de diagnosi, des del disseny de les proves de detecció de falles, fins a les etapes d'aïllament quantitatiu i identificació de falles.

Així, la metodologia inclou algoritmes per obtenir de manera automàtica l'expressió simbòlica dels generadors de residus millorant l'aïllabilitat estructural de les falles. Aquests algoritmes es basen en el model estructural del sistema.

Les etapes de detecció, aïllament, i identificació de falles es representen com problemes de satisfacció de restriccions en dominis continus i es resolen per mitjà de tècniques de consistència basades en intervals. Una millora en l'aïllament qualitatiu de les falles s'obté per raonament amb els signes dels símptomes, que s'obtenen de relacions de redundància analítica o de models *bond graph* del sistema. L'aïllament quantitatiu i l'identificació de les falles es realitza per mitjà de l'estimació dels paràmetres que fallen.

Aquesta tesi també presenta una anàlisi empírica inicial de les diferències entre les tècniques basades en intervals i les basades en tècniques

estadístiques. Els mètodes inclosos en aquesta comparació són les tècniques ICT per una banda i per una altra el filtre estès de Kalman i la tècnica de decisió estadística *Z-test*.

Les prestacions i l'eficiència de les contribucions de la tesi s'il·lustren a través d'uns quants exemples d'aplicació, que cobreixen diferents nivells de complexitat.

# Notation and Abbreviations

## Notation

### Structural Analysis

$M$	A diagnosis model
$X$	The set of unknown variables
$Z$	The set of known variables
$c$	A constraint
$v$	A variable
$\mathcal{M}$	A matching
$\mathcal{E}$	A set of edges
$e$	An edge
$\mathcal{C}_{MSO}$	A collection of MSO sets
$\varphi$	The structural redundancy of a model
$u$	An input variable
$y$	An output variable
$p^{(i,j)}$	A path through the structure graph from input $u_i$ to output $y_j$
$C$	A condition associated to a set of equations of the model

### Fault detection, isolation and identification based on Interval Models

Scalars are represented by italics, vectors by bold lower-case letters, and intervals by upper-case letters.



$\wedge$	(superscript) estimate
$\sim$	(superscript) measurement
$\mathcal{V}$	A set of numeric variables
$\mathcal{D}$	A set of domains
$\mathcal{C}$	A set of constraints
$x$	A variable/a state variable
$y$	An output variable
$u$	An input variable
$\theta$	A parameter

## Abbreviations

BG	Bond Graph
CBR	Case-Based Reasoning
DWT	Discrete Wavelet Transform
EKF	Extended Kalman Filter
FA	False Alarm
FD	Fault Detection
FDI	Fault Detection and Isolation
FDII	Fault Detection, Isolation, and Identification
FSM	Fault Signature Matrix
GDE	General Diagnosis Engine
IA	Interval Analysis
ICT	Interval-based Consistency Technique
KF	Kalman Filter
MA	Missed Alarm
MBD	Model-based Diagnosis
MIA	Modal Interval Analysis
MSO	Minimal Structurally Overdetermined
SO	Structurally Overdetermined
TCG	Temporal Causal Graph
WT	Wavelet Transform

# Contents

Notation and Abbreviations	xi
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	2
1.2 Outline and contributions . . . . .	3
1.2.1 Publications . . . . .	5
<b>2 Related Work in Fault Diagnosis</b>	<b>9</b>
Summary . . . . .	9
2.1 Fault Definitions . . . . .	9
2.2 Diagnosis Systems . . . . .	12
2.3 Research within Model-based Diagnosis . . . . .	16
2.3.1 DX Community . . . . .	17
2.3.2 FDI Community . . . . .	20
2.4 Robustness in MBD . . . . .	23
2.4.1 Bounding Approaches . . . . .	25
2.4.2 Robustness by using Quantitative, Qualitative, and Semi- qualitative Information . . . . .	25
2.4.3 Interval Techniques used for Fault Detection . . . . .	27
<b>3 Residual Generation</b>	<b>29</b>
Summary . . . . .	29
3.1 Related Work . . . . .	30
3.2 Structural Analysis . . . . .	32
3.3 Matching Algorithms . . . . .	35
3.4 A Residual Generation Algorithm . . . . .	40

3.4.1	Computational Complexity . . . . .	43
3.5	Causal Interpretations . . . . .	44
3.6	Structural Detectability and Isolability Analysis . . . . .	48
3.7	Active Structural Isolation . . . . .	50
3.7.1	Algorithms . . . . .	51
3.8	A Case of Multiple Modes of Operation . . . . .	54
3.9	Conclusions and Contributions . . . . .	57
<b>4</b>	<b>Robust Fault Detection based on Interval Models</b>	<b>59</b>
	Summary . . . . .	59
4.1	Related Work . . . . .	60
4.2	A Tool based on Modal Intervals: SQualTrack . . . . .	62
4.2.1	Dynamic Refinement of the Parameters Space . . . . .	66
4.2.2	Dynamically Refining the Measurements Space . . . . .	69
4.3	Fault Detection as a Constraint Satisfaction Problem . . . . .	71
4.3.1	Interval-based Consistency Techniques . . . . .	74
4.3.1.1	Interval Observer . . . . .	76
4.4	Comparing Interval-based Consistency Techniques and Other Methods for Fault Detection . . . . .	78
4.4.1	ICTs vs. SQualTrack . . . . .	78
4.4.1.1	Application Example . . . . .	79
4.4.2	ICTs vs. a Statistical Decision: Extended Kalman Filter and Z-test . . . . .	85
4.4.2.1	Summary of Main Differences . . . . .	88
4.4.2.2	Application Example . . . . .	88
4.5	Conclusions and Contributions . . . . .	99
<b>5</b>	<b>Qualitative Fault Diagnosis: Isolation</b>	<b>101</b>
	Summary . . . . .	101
5.1	Related Work . . . . .	102
5.2	Additional Information Sources for Fault Isolation . . . . .	103
5.3	Signs of the Symptoms using the Sensitivity Matrix . . . . .	104
5.3.1	Qualitative Fault Isolation Results using SQualTrack . . . . .	106
5.3.2	An Application to Coupled Water Tanks . . . . .	107

5.3.2.1	Faults . . . . .	107
5.3.2.2	FSM taking into account Residual Signs . . . . .	108
5.3.3	Diagnosis Results . . . . .	110
5.3.4	Simulation Results . . . . .	110
5.4	Signs of the Symptoms using Bond Graph Models . . . . .	113
5.4.1	Faults with a Discontinuous Change in a Measurement . . . . .	117
5.4.2	Symbol Generation using Z-test and Wavelets . . . . .	122
5.5	Conclusions and Contributions . . . . .	125
<b>6</b>	<b>Quantitative Fault Diagnosis: Isolation and Identification</b>	<b>127</b>
	Summary . . . . .	127
6.1	Related Work . . . . .	127
6.2	Hypothesis Tests . . . . .	129
6.2.1	Algorithm for Fault Isolation and Identification . . . . .	130
6.3	An Application To Coupled Water Tanks . . . . .	131
6.3.1	Simulation Results . . . . .	133
6.4	Conclusions and Contributions . . . . .	136
<b>7</b>	<b>Applications</b>	<b>137</b>
	Summary . . . . .	137
7.1	Positioning Control System of an Offshore Vessel . . . . .	138
7.1.1	Active Structural Isolation . . . . .	141
7.2	Applications in Electrical Distribution Systems . . . . .	142
7.2.1	Nine Bus Case Network . . . . .	142
7.2.2	Three-phase $\Delta - Y$ Coupled Transformer . . . . .	144
7.3	An Automotive Engine . . . . .	146
7.3.1	Structural Analysis . . . . .	146
7.3.2	Fault Detection in the Air Intake System . . . . .	149
7.3.3	Experimental Results: Detection . . . . .	151
7.3.3.1	First Scenario . . . . .	151
7.3.3.2	Second Scenario . . . . .	152
7.3.4	Diagnosis: Signs of the Symptoms . . . . .	154
7.4	DAMADICS Benchmark . . . . .	156
7.4.1	Residual Generation . . . . .	157

## CONTENTS

---

7.4.2	Fault Detection . . . . .	160
7.4.3	Diagnosis Results . . . . .	162
7.4.4	Simulation Results . . . . .	162
7.5	Alcoholic Fermentation Process . . . . .	165
7.5.1	System Description . . . . .	165
7.5.2	Fault Detection Results . . . . .	167
7.5.3	Fault Isolation Results . . . . .	167
7.5.4	Isolation Time . . . . .	170
<b>8</b>	<b>Conclusions and Future Work</b>	<b>171</b>
	<b>References</b>	<b>177</b>

# List of Tables

3.1	Incidence matrix of Example 3.2.1. . . . .	33
3.2	Incidence matrix with unknown variables showing one complete matching. . . . .	39
3.3	Dependency table for the example. . . . .	46
3.4	Detectability and isolability in a single fault case. . . . .	49
3.5	Output reachability from $u_1$ . . . . .	53
3.6	Output reachability from $u_2$ . . . . .	53
3.7	Active structural detectability and isolability of some single faults. . . . .	53
3.8	Relations between the conditions and the operation modes. . . . .	56
3.9	MSO sets of operation modes 1 and 2. . . . .	56
3.10	Detectability and isolability using both operation modes. . . . .	57
4.1	Parameter intervals of the chemical plant example. . . . .	68
4.2	Elementary analytical relations of the two coupled tanks system . . . . .	81
4.3	MSO sets of the two-tank system. . . . .	81
4.4	Main differences between both techniques. . . . .	88
4.5	Fault Detection system design parameters. . . . .	92
4.6	Results with different fault magnitudes and noise level equal to 1.5%. . . . .	94
4.7	Results with different fault magnitudes and noise level equal to 3%. . . . .	95
5.1	Extended model including faults for the two coupled tanks system . . . . .	108
5.2	Influence of the faults in ARRs. . . . .	109
5.3	Sign of the sensitivity of a symptom with respect to each fault. . . . .	109
5.4	ARRs and their related fault modes using the sign of the symptom. . . . .	110
5.5	Diagnosis using and not using signs. . . . .	110

---

**LIST OF TABLES**

5.6	Diagnostics for the fault scenario . . . . .	112
5.7	Fault signatures when $f_2$ is measured. . . . .	122
6.1	Elementary analytical relations of the two coupled tanks system .	132
6.2	Fault signature matrix. . . . .	133
7.1	Application examples. . . . .	137
7.2	Structural model for a DP vessel of AUTS class. . . . .	139
7.3	Detectability and isolability of single faults. . . . .	140
7.4	Output reachability from $u_i$ . . . . .	141
7.5	Incidence matrix for the system represented with phase compo- nents in composite notation. . . . .	144
7.6	Detectability and isolability in a single fault case. . . . .	145
7.7	Measurable variables. . . . .	147
7.8	Results of Algorithm 5 applied to the application example . . . .	148
7.9	Incidence Matrix of the valve. . . . .	157
7.10	MSO sets of the application example. . . . .	158
7.11	Results of Algorithm 2 applied to the application example . . . .	159
7.12	Fault isolability analysis matrix. . . . .	159
7.13	Fault isolability analysis matrix using the 29 MSO sets. . . . .	160
7.14	Fault signature matrix . . . . .	161
7.15	ARRs and their related fault modes using the sign of the symptom	162
7.16	Diagnosis using and not using signs . . . . .	163
7.17	Diagnostics for the $f_{13}$ fault scenario . . . . .	163
7.18	Parameters and yield coefficients. . . . .	167
7.19	Fault detection results . . . . .	167
7.20	Fault isolation results . . . . .	168
7.21	Scenario(i). Fault isolation results . . . . .	169
7.22	Scenario(ii). Fault isolation results . . . . .	170

# List of Figures

1.1	Architecture of the diagnosis system. . . . .	3
2.1	Time dependency of faults: (a) abrupt; (b) incipient; (c) intermittent. . . . .	11
2.2	Classification of diagnosis methods. . . . .	15
2.3	Basic Diagnosis Scheme. . . . .	18
2.4	The three tasks of a model-based fault diagnosis system. . . . .	21
2.5	Two-stage structure of model-based FDI procedure. . . . .	21
3.1	Structure graph of Example 3.2.1. . . . .	33
3.2	Bi-partite graph of Example 3.2.1. . . . .	34
3.3	A complete matching represented on a bi-partite graph. . . . .	39
3.4	Order in which the matched edges are expanded. . . . .	40
3.5	Oriented graphs showing the order in which the unknown variables can be determined in (a) $MSO_1$ , (a) $MSO_2$ , (c) $MSO_3$ and (d) $MSO_4$ . . . . .	47
3.6	Structure graph of Example 3.7.1. . . . .	53
3.7	An electric circuit with multiple operation modes. . . . .	55
4.1	The three zones defined by the internal and external estimates of the exact envelope. . . . .	63
4.2	Procedure used to find the new bounds of one parameter. . . . .	67
4.3	Scheme of the chemical tank equipped with a heat exchanger . . .	67
4.4	A fault in pump $B1$ using SQualTrack, without the refinement of the parameters space. . . . .	69



**LIST OF FIGURES**

---

4.5	A fault in pump <i>B1</i> using SQualTrack, with the refinement of the parameters space. . . . .	70
4.6	A fault in pump <i>B1</i> using SQualTrack, without the refinement of the measurements space. . . . .	71
4.7	A fault in pump <i>B1</i> using SQualTrack, with the refinement of the measurements space. . . . .	72
4.8	Box solution of Example 4.3.1. . . . .	76
4.9	Consistency test of the two interval approaches used in this thesis. . . . .	79
4.10	Diagram of the coupled water tanks system. . . . .	80
4.11	Weak-3B Consistency Fault Detection using $ARR_1$ . The fault is detected from sample 408. . . . .	82
4.12	SQualTrack Fault Detection using $ARR_1$ . The fault is not detected. . . . .	83
4.13	Weak-3B Consistency Fault Detection using $ARR_2$ . The fault is detected from sample 410. . . . .	83
4.14	SQualTrack Fault Detection using $ARR_2$ . The fault is not detected. . . . .	84
4.15	Block diagram of the observer-based fault diagnosis approach. . . . .	85
4.16	Scheme of the fault detection using the Z-test. . . . .	86
4.17	Confidence level defined by $\alpha$ . . . . .	87
4.18	The two tank system schematic. . . . .	89
4.19	EKF and Z-test fault detection. $N_2 = 9$ , $z^+ = 3.0$ , and $Q = [0.00003]I^2$ . . . . .	93
4.20	Interval-based consistency technique fault detection. $\omega = 25$ , and $V_i = \pm 3.0\sigma$ . . . . .	93
4.21	False alarm rate using the interval-based technique. Incipient clogging fault in the output pipe <i>P2</i> . . . . .	96
4.22	Fault detection times using the interval-based technique. Incipient clogging fault in the output pipe <i>P2</i> . . . . .	96
4.23	Missed detection rate (dark surface) and False alarm rate (light surface) using the interval-based technique. Abrupt clogging fault in the output pipe <i>P2</i> . . . . .	97
4.24	Fault detection times with different noise levels using the interval-based technique: 1.5% solid line and 3% dashed line. Incipient clogging fault in the output pipe <i>P2</i> . . . . .	97

**LIST OF FIGURES**

---

4.25	Fault detection times using a statistical decision approach. Incipient clogging fault in the output pipe $P2$ . . . . .	98
4.26	False alarm rate using a statistical decision approach. Incipient clogging fault in the output pipe $P2$ . . . . .	98
5.1	SQualTrack Fault Detection: (a) when the sign of the symptom is $+1$ , and (b) when the sign of the symptom is $-1$ . . . . .	106
5.2	SQualTrack Fault Detection using $r_1$ corresponding to fault $f_7$ beginning at sample 200. The fault is detected from sample 221. . . . .	111
5.3	SQualTrack Fault Detection using $r_2$ corresponding to fault $f_7$ beginning at sample 200. The fault is detected from sample 214. . . . .	112
5.4	One-tank system and its bond graph model and temporal causal graph. . . . .	116
5.5	Forward propagation for $R_1^+$ fault. . . . .	116
5.6	Slope of the residual for a one-tank system, function of $f$ and $h_2$ . . . . .	118
5.7	Nominal (black solid line) and faulty (red dashed line) behavior for $h_2$ when the tank is filling. $C_{t1}^-$ (20% decrease) is injected at 20 s. $(++)$ signature. . . . .	118
5.8	Nominal and faulty behavior for $h_2$ when the tank is filling. $C_{t1}^-$ (20% decrease) is injected at 50 s. $(+-)$ signature. . . . .	119
5.9	Nominal and faulty behavior for $h_2$ when the tank is emptying. $C_{t1}^-$ (20% decrease) is injected at 20 s. $(+-)$ signature. . . . .	119
5.10	Nominal and faulty behavior for $f_3$ when the tank is filling. $R_1^+$ (20% increase) is injected at 20 s. $(--)$ signature. . . . .	120
5.11	Nominal and faulty behavior for $f_3$ when the tank is filling. $R_1^+$ (20% increase) is injected at 100 s. $(-+)$ signature. . . . .	120
5.12	Mass-spring-damper system. . . . .	121
5.13	Temporal causal graph of the mass-spring-damper system. . . . .	121
5.14	Nominal and faulty behavior for $f_2$ when $m^+$ is injected at different times. . . . .	123
5.15	DWT applied to a measurement with a non-discontinuous change when a fault occurs. . . . .	124

## LIST OF FIGURES

---

5.16 DWT applied to a measurement with a discontinuous change when a fault occurs. . . . .	124
6.1 Measured signals when a clogging fault in the output pipe of $T1$ is present. . . . .	134
6.2 Fault identification of parameter $f_1$ . . . . .	135
6.3 Fault identification of parameter $f_5$ . . . . .	136
7.1 Structure graph for the application example. . . . .	140
7.2 Single-line diagram of the nine bus case network. . . . .	143
7.3 A schematic figure of the turbo-charged engine. . . . .	146
7.4 Incidence matrix of the engine structural model. . . . .	148
7.5 First scenario fault detection. Top: no fault. Bottom: gain-fault in the sensor of pressure $p_{ic}$ beginning at sample 600. The fault is detected from sample 604. . . . .	152
7.6 First scenario without faults. The upper plot shows measured and estimated manifold pressure. . . . .	152
7.7 Second scenario fault detection. Top: no fault. Bottom: gain-fault in the sensor of pressure $p_{im}$ . The fault is detected at the same time as the fault appears. . . . .	153
7.8 Second scenario without faults. The upper plot shows measured and estimated manifold pressure. . . . .	153
7.9 First scenario with a gain-fault in the sensor of pressure $p_{ic}$ . Starts at 600, detected at 604. . . . .	155
7.10 Second scenario with a gain-fault in the sensor of pressure $p_{im}$ . Starts at 800, detected at the same time as the fault. . . . .	155
7.11 General scheme of the DAMADICS valve. . . . .	156
7.12 SQualTrack Fault Detection using $r_1$ corresponding to fault $f_{13}$ beginning at 900 s. The fault is detected from 915 s. . . . .	164
7.13 SQualTrack Fault Detection using $r_2$ corresponding to fault $f_{13}$ beginning at 900 s. The fault is detected from 903 s. . . . .	164
7.14 Fault detection. Scenario (i), fault is on parameter $\mu_m$ . . . . .	168
7.15 Fault detection. Scenario (ii), fault is on parameter $K_s$ . . . . .	168

## LIST OF FIGURES

---

8.1	Architecture of the diagnosis system. . . . .	172
-----	---	-----



# Chapter 1

## Introduction

Nowadays the industrial processes are characterized by a high and increasing level of automation and technological complexity. Those characteristics have given a critical role to the systems supervision tasks, because of the need to anticipate, prevent, detect, and ultimately correct the behavior of the system when it deviates from its normal expected behavior. In general sense, this kind of deviation from the acceptable, usual, standard condition is called a fault [Isermann & Ballé \(1997\)](#).

Faults in process components of industrial plants may have serious impacts on production, such as increment of operational costs, decrease of the production quality, etc. Even more serious could be the consequences, for example on the people or on the environment, of an accident because of plant operation in presence of faults.

To increase the system reliability and safety, there is a need to detect and identify the nature of the fault, and hence, fault diagnosis systems have become more and more important. Tasks related to fault diagnosis are assigned to the supervision computers, and a desirable goal of these tasks is to detect faults early to, for example, initiate recovery actions. Early diagnosis can avoid, in some cases, big catastrophes (ships, airplanes, great structures) and in other cases, can reduce the maintenance costs. There are many research works related to fault diagnosis and their results are now applied to continuous processes of production, where the availability of the machinery is critical from an economic point of view.

A branch within fault diagnosis research area is the model-based one. Model-based diagnosis methods are based on a set of assumptions, one of them is that the mathematical model represents accurately the system dynamics. But, an accurate mathematical description of the process is never available. In the same way, as the complexity of a dynamic system increases, the system and its disturbances modeling task becomes more difficult. As a consequence of such difficulties, model-based diagnosis methods must be robust to model errors and other sources of uncertainty, e.g. the uncertainty associated with the measurements, and so, the methods must be only sensitive to faults. In order to handle the problems due to the uncertainty, some approaches, like the ones proposed in this thesis, use interval models.

## 1.1 Motivation

The starting point of the research done in this thesis was SQualTrack [Armengol \(1999\)](#), a software package for fault detection based on Modal Interval Analysis (MIA). The initial research objective was to analyze the capabilities of this software, and to improve its fault detection performance. In this regard, some alternatives of improvement are proposed. Due to some limitations of SQualTrack, e.g. it can only be applied to single-equation residual generators, and in order to include interval-based proposals for fault diagnosis, isolation and identification, other interval-based approaches are studied.

Thus, the main objective of this thesis becomes: to propose a methodology for fault detection, isolation and identification based on interval models. This methodology should include both off-line and online stages within the supervision process. For instance, the fault detection in complex systems requires a previous off-line stage, which is the obtention of appropriate residual generators.

In this sense, this thesis includes some algorithms to obtain in an automatic way the symbolic expression of the residual generators enhancing the structural isolability of the faults. Fault detection, isolation, and identification are tackled by means of interval-based techniques to take into account the uncertainty in the systems.

## 1.2 Outline and contributions

Regarding an architecture of a supervision system like the one presented in Fig. 1.1, the scope of this thesis includes contributions in each stage of the supervision system. The outline of this thesis follows these stages.

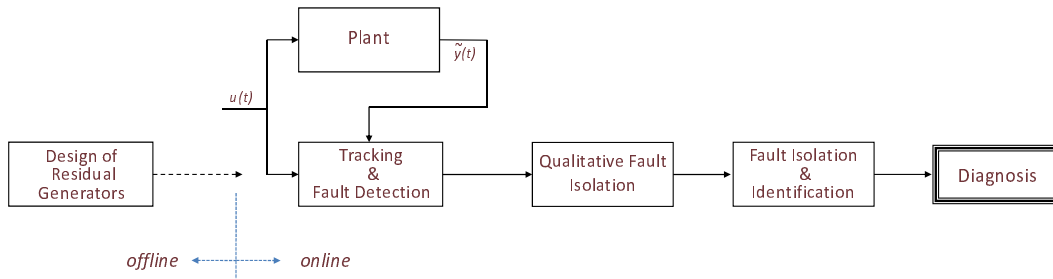


Figure 1.1: Architecture of the diagnosis system.

Chapter 2 gives an overview of the fault diagnosis techniques. From the wide variety of techniques, it is focused on model-based ones which use an explicit model of the system to be diagnosed. The robustness problem is presented, i.e. the ability to distinguish between faults and several sources of uncertainty, such as modeling or measurement errors. In particular, bounded approaches, i.e. approaches which use interval models, are introduced.

In Chapter 3, a *residual generation* toolbox based on structural analysis and developed in Matlab<sup>®</sup> is proposed. The toolbox includes algorithms to find automatically all minimal structurally overdetermined (MSO) sets in a structural model of a system, all causal models for each MSO set, and routines to represent them and to analyze the detectability and isolability of faults. Moreover, it includes algorithms for investigation of the possibilities of active structural isolation to enhance the structural isolability of faults.

Contributions related to the stage of *tracking and fault detection* in Fig. 1.1, are presented in Chapter 4. During the development of this thesis, mainly two interval approaches were used for fault detection:

- As a starting point, a software package called SQualTrack based on Modal Interval Analysis. In this case, two improvements are proposed to increase the fault detection performance of this software.



- Interval-based Consistency Techniques - ICTs, a combination between “classic” interval analysis and constraint propagation techniques. Concerning the ICTs, the work developed during this thesis includes: (i) state the fault detection problem as a constraint satisfaction problem, and (ii) design and develop a fault detection system using the solver called RealPaver [Granvilliers & Benhamou \(2006\)](#).

Chapter 4 also includes an empirical comparison between the performance of the above mentioned approaches and between ICTs and a statistical-decision technique: Extended Kalman filter and Z-test.

Proposed approaches relative to the *qualitative fault isolation* stage in Fig. 1.1, are described in Chapter 5. The fault isolation task is improved by using additional information obtained from the fault models and the effect of faults on the system. Two different alternatives, that take into account qualitative information from deviation in symptoms, are assessed. First, a method that uses the internal form of the consistency relations is proposed to complete the fault signature matrix. Second, a method based on bond graph models that derives the qualitative information from a temporal causal graph is analyzed and extended. This extension includes the analysis of faults that cause a discontinuous change in a measurement.

Qualitative methods for fault isolation are very useful but can not discriminate faults that show no qualitative differences, or even more, sometimes the qualitative effect is indeterminate. In Chapter 6, interval-based consistency techniques are used to improve the *fault isolation* and perform the *fault identification*. Basically, this stage is done by refining the fault hypothesis set, and estimating the magnitude of the fault.

The methods and the theory developed in the previous chapters are applied to several examples in Chapter 7. The application examples include:

- Positioning control system of an offshore vessel
- Electrical distribution systems
- Automotive engine

- DAMADICS Benchmark
- Alcoholic Fermentation Process

Finally, Chapter 8 provides some conclusions and orientations for future work.

### 1.2.1 Publications

During the research work leading to this thesis, the following conference and journal papers have been published:

- Acosta, G., Gelso, E.R. & Verucchi, C. (2004). Detección en línea de fallos de aislamiento en devanados de máquinas de inducción. In *XIX Congreso Argentino de Control Automático AADECA '2004*, Buenos Aires, Argentina.
- Prieto, O.J., Gelso, E.R., Pulido, B., Rodríguez, J.J. & Maestro, J.A. (2004). Specification for a versatile data acquisition module for a supervision application. In *International Workshop of Agents and Multi-agents Systems IWPAAMS'2004*, Burgos, Spain.
- Pulido, B., Rodríguez, J.J., Alonso-González, C., Prieto, O.J. & Gelso, E.R. (2004). Naive bayesian classifier for fault identification in continuous processes. In *International Workshop of Agents and Multi-agents Systems IWPAAMS'2004*, Burgos, Spain.
- Pulido, B. & Gelso, E.R. (2005). Viabilidad de una técnica de compilación de dependencias para diagnosis basada en modelos en tareas de supervisión. In *XI Conferencia de la Asociación Española para la Inteligencia Artificial, CAEPIA '05*, Santiago de Compostela, Spain.
- Pulido, B., Rodríguez, J.J., Alonso, C., Prieto, O.J. & Gelso, E.R. (2005). Diagnosis of continuous dynamic systems: Integrating consistency based diagnosis with machine learning techniques. In *16th IFAC World Congress*, Prague, Czech Republic.

- Acosta, G.G., Verucchi, C.J. & Gelso, E.R. (2006). A current monitoring system for diagnosing electrical failures in induction motors. *Mechanical Systems and Signal Processing*, **20**, 953–965.
- Calderón-Espinoza, G., Armengol, J., Vehí, J. & Gelso, E.R. (2007). Dynamic diagnosis based on interval analytical redundancy relations and signs of the symptoms. *AI Communications*, **20**, 39–47.
- Castillo, S.M., Gelso, E.R. & Armengol, J. (2007a). Modelos con incertidumbre: Identificación de parámetros intervalares y simulación usando cuantificadores. In *II Congreso Español de Informática (CEDI'07), I Simposio en Modelado y Simulación de Sistemas Dinámicos*, Zaragoza, Spain.
- Castillo, S.M., Gelso, E.R., Calm, R. & Armengol, J. (2007b). Estimación de parámetros intervalares para la detección robusta de fallos. In *IX Jornadas de ARCA. Sistemas Cualitativos y Diagnóstico*, 41–48, Lloret de Mar, Spain.
- Gelso, E.R., Castillo, S.M. & Armengol, J. (2007a). Robust fault detection using consistency techniques for uncertainty handling. In *IEEE International Symposium on Intelligent Signal Processing, WISP 2007*, 77–82, Alcalá de Henares, Spain.
- Gelso, E.R., Castillo, S.M. & Armengol, J. (2007b). Construyendo una plataforma para detección y diagnóstico de sistemas dinámicos basada en análisis intervalar modal y redundancia analítica. In *IX Jornadas de ARCA. Sistemas Cualitativos y Diagnóstico*, 15–21, Lloret de Mar, Spain.
- Gelso, E.R., Castillo, S.M. & Armengol, J. (2007c). Diagnosis based on interval analytical redundancy relations and signs of the symptoms. In *IFAC International Workshop on Intelligent Manufacturing Systems IMS'07*, Alicante, Spain.
- Khosravi, A., Armengol, J. & Gelso, E.R. (2007). An interval intelligent-based approach for fault detection and modelling. In *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2007)*, London, England.

- Castillo, S.M., Gelso, E.R. & Armengol, J. (2008). Constraint satisfaction techniques under uncertain conditions for fault diagnosis in nonlinear dynamic systems. In *16th Mediterranean Conference on Control and Automation*, Ajaccio-Corsica, France.
- Gelso, E.R., Biswas, G., Castillo, S.M. & Armengol, J. (2008a). A comparison of two methods for fault detection: a statistical decision, and an interval-based approach. In *19th International Workshop on Principles of Diagnosis DX*, 261–268, Blue Mountains, Australia.
- Gelso, E.R., Castillo, S.M. & Armengol, J. (2008b). Fault diagnosis by refining the parameter uncertainty space of nonlinear dynamic systems. In *American Control Conference (ACC08)*, Seattle, USA.
- Gelso, E.R., Castillo, S.M. & Armengol, J. (2008c). *Frontiers in Artificial Intelligence and Applications - Artificial Intelligence Research and Development (CCIA'08)*, chap. An interval-based approach for fault isolation and identification in continuous dynamic systems, 421–429. IOS Press.
- Gelso, E.R., Castillo, S.M. & Armengol, J. (2008d). *Frontiers in Artificial Intelligence and Applications - Artificial Intelligence Research and Development (CCIA'08)*, chap. An algorithm based on structural analysis for model-based fault diagnosis, 138–147. IOS Press.
- Gelso, E.R., Frisk, E. & Armengol, J. (2008e). Robust fault detection using consistency techniques with application to an automotive engine. In *17th IFAC World Congress*, 5512–5517, Seoul, Korea.
- Armengol, J., Vehí, J., Sainz, M.Á., Herrero, P. & Gelso, E.R. (2009b). SQualTrack: a tool for robust fault detection. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, **39**, 475–488.
- Armengol, J., Bregón, A., Escobet, T., Gelso, E.R., Krysanter, M., Nyberg, M., Olive, X., Pulido, B. & Travé-Massuyès, L. (2009a). Minimal structurally overdetermined sets for residual generation: A comparison of

## 1.2 Outline and contributions

---

alternative approaches. In *7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes SAFEPROCESS*, Accepted, Barcelona, Spain.

- Gelso, E.R. & Blanke, M. (2009). Structural analysis extended with active fault isolation – methods and algorithms. In *7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes SAFEPROCESS*, Accepted, Barcelona, Spain.

# Chapter 2

## Related Work in Fault Diagnosis

### Summary

This chapter reviews the basic definitions and state of the art of fault diagnosis techniques. From the wide variety of techniques, it is focused on model-based ones which use an explicit model of the system to be diagnosed. This chapter introduces them from the point of view of two research communities: the FDI (Fault Detection and Isolation) community, formed by researchers with a background in control systems engineering, and the DX (Principles of Diagnosis) community, formed by researchers with a background in computer science and intelligent systems. The robustness problem is presented, i.e. the ability to distinguish between faults and modeling uncertainty. In particular, bounded approaches, i.e. approaches which use interval models, are introduced.

### 2.1 Fault Definitions

The definition of the term fault exists in the literature in different ways. In the context of this thesis, the Longman Dictionary [Pearson Longman \(2008\)](#) defines fault as “something wrong with something: **a)** something that is wrong with a machine, system, design etc, which prevents it from working properly; **b)** something that is wrong with something, which could be improved; **c)** a mistake in the way that something was made, which spoils its appearance”. According to [Frank \(1990\)](#), a fault is defined as any kind of malfunction in the actual dynamic system, the plant, that leads to an unacceptable anomaly in the overall system

performance. Such malfunctions may occur either in the sensors (instruments), or actuators, or in the components of the process.

The IFAC, the International Federation of Automatic Control, Technical Committee on Fault Detection, Supervision and Safety of Technical Processes (SAFE-PROCESS) has made an effort to come to accepted definitions, and some definitions can be found, for example in [Isermann \(2006\)](#); [Isermann & Ballé \(1997\)](#). In those works, a fault is defined as:

**Definition 2.1.1.** *A fault is an unpermitted deviation of at least one characteristic property or parameter of the system from the acceptable, usual, standard condition.*

Taking into account the mentioned references, in this thesis the term fault differs from the terms failure and malfunction. In contrast to the term fault, the notion of a failure suggests a complete breakdown of a system or a component, and the inability of it to accomplish its function.

**Definition 2.1.2.** *A failure is a permanent interruption of a system's ability to perform a required function under specified operating conditions.*

**Definition 2.1.3.** *A malfunction is an intermittent irregularity in the fulfillment of a system's desired function.*

According to the time dependency of faults, the faults can be classified in three categories, as shown in Fig. 2.1, [Isermann \(2005\)](#).

**Abrupt faults** Stepwise, i.e. the fault occurs abruptly and then stays present.

An abrupt fault could be, for example, a sudden connection cut-off of a wire in an electrical circuit.

**Incipient faults** Driftlike, i.e. a fault that gradually increases in size. An incipient fault could be, for example, the accumulation of sediment in pipes of hydraulic systems.

**Intermittent faults** A fault that occurs and disappears repeatedly. A typical example of an intermittent fault is caused by bad electrical contacts, e.g. faulty relays.

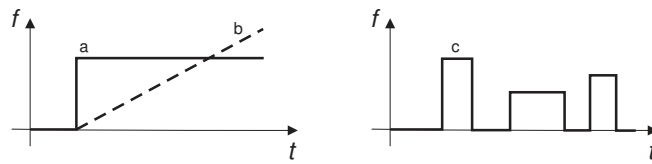


Figure 2.1: Time dependency of faults: (a) abrupt; (b) incipient; (c) intermittent.

Since a plant can be split into three kinds of subsystems, i.e. actuators, the process, and sensors, such a decomposition leads directly to three classes of faults [Witczak \(2003\)](#):

**Actuators faults**, which can be viewed as any malfunction of the equipment that actuates the system, e.g. a blockage in a valve.

**Process faults** (or component faults), which cause changes within the process, e.g. a leak in a tank in a two-tank system.

**Sensors faults**, which can be viewed as serious measurements variations. For example, they are offsets or drifts of sensors.

The faults can be further classified into one of the following categories [Gertler \(1998\)](#):

**Additive faults:** These are unknown inputs acting on the plant, which are normally zero and which, when present, cause a change in the plant outputs independent of the known inputs. Such faults best describe plant leaks, loads, etc.

**Multiplicative faults:** These are changes (abrupt or gradual) in some plant parameters. They cause changes in the plant outputs which depend also on the magnitude of the known inputs. Such faults best describe the deterioration of the plant equipment, such as surface contamination, clogging, or the partial or total loss of power.



## 2.2 Diagnosis Systems

A diagnosis system is one that is capable of identifying the nature of a problem by examining the observed symptoms [Balakrishnan & Honavar \(1998\)](#). The output of the system is a diagnosis (and possibly an explanation or justification of it).

Typically a diagnosis system is provided with a set of symptoms (observations or measurements) as input. The system's task is to identify a probable cause that could explain the observed symptoms. These systems are needed to satisfy the following requirements [Patton \*et al.\* \(2000\)](#):

- Early detection of small faults with abrupt or incipient time behavior.
- Diagnosis of actuator, sensor or process component faults.
- Detection of faults in closed-loop systems.
- Supervision of processes undergoing transient variations.

In particular, the goal for early diagnosis is to have enough time for counteractions like other operations, reconfiguration, maintenance or repair.

There are several ways in which the different diagnosis approaches can be classified and some of them are described below.

In [Balakrishnan & Honavar \(1998\)](#), the diagnosis approaches are distinguished based on (i) the way in which the diagnosis system comes to know of the relationship between the observed symptoms and the consequent diagnosis, (ii) the way of representing this relationship, and (iii), how the system uses this representation for diagnosing faults. Therefore, the approaches are classified in model-based systems, knowledge-based systems, case-based reasoning systems, and machine learning techniques.

- Model-based systems. The diagnostic knowledge is stored as an explicit model of the domain. When the system's observed behavior conflicts with the system's expected behavior, the diagnosis task is to identify the (faulty) system component(s) that explain the anomaly. This approach is useful in domains where the underlying physical principles are largely known (e.g., in artificial systems). Although model-based systems have the ability to

produce accurate diagnosis, they have several drawbacks which make them inapplicable in many real world contexts. Their need for an accurate model of the domain may not be possible to satisfy in many cases.

- Knowledge-based systems. The knowledge (gained through experience) of human experts is codified in a knowledge-based system. They are useful, for instance, in many practical scenarios in which precise models of the domain may be unavailable. Expert systems are an application example of these systems. They are programs that model the expertise (knowledge) and reasoning capabilities of qualified specialists within fairly narrow domains. They are typically composed of three essential modules: a knowledge base that captures the expertise of the specialist in some appropriate form (e.g. rules), an inference engine that mimics the specialist's reasoning process, and a working memory to hold the facts provided by the user (e.g. symptoms in a diagnosis task) and intermediate conclusions derived by the inference procedure. The efficacy of this approach depends on the faithfulness of the encoding as well as the quality of the expert's domain experience.
- Case-based reasoning systems. In case-based reasoning (CBR) systems knowledge is stored in the form of cases, where a case can be thought of as a situation that was experienced in the past and resulted in some relevant action. The cases are stored in a library, indexed appropriately to facilitate efficient retrieval of the cases. Given a current experience or situation, the attributes of the input are used to index into the case library and retrieve the best matching case (or set of cases) according to some suitably defined matching criterion. This approach attempts to circumvent the difficult task of extracting and codifying the domain knowledge of the expert. Unlike model-based systems and expert systems, case-based systems neither model the domain knowledge nor the diagnostic reasoning of the domain expert. Instead, the knowledge is implicitly represented in the repository of cases. The drawbacks of CBR systems include the high computational cost associated with the matching procedure and the storage cost associated with the organization of the case library.

- Machine learning techniques. A learning system is essentially one that is capable of improving its performance at a given task (or a set of tasks) through experience. They extract knowledge that is implicitly provided by a large number of examples. These techniques, for example, are based on artificial neural networks, statistical pattern classification systems, decision trees, etc.

In [Gertler \(1998\)](#), fault diagnosis methods are classified into two major groups: those which do not utilize the mathematical model of the plant and those which do. According to this classification, the model-free methods range from physical redundancy and special sensors through limit-checking and spectrum analysis to logical reasoning.

- Physical redundancy (or hardware redundancy). In this approach, multiple sensors are installed to measure the same physical quantity. Typically, a voting technique is applied to the hardware redundant system to decide if a fault has occurred and its location among all the redundant system components [Simani \*et al.\* \(2003\)](#). There are at least three problems associated with the use of physical redundancy: extra hardware cost, it requires space, and adds weight to the system [Nyberg \(1999\)](#).
- Limit checking. In this approach, plant measurements are compared to preset limits. Exceeding the threshold indicates a fault situation. This approach can suffer from the following drawbacks: (i) multitude of alarms and difficult isolation when, for instance, the effect of a single component fault propagates to many plant variables, and (ii), difficult or quite conservative setting of the thresholds due to normal input variations.

According to [Samantaray & Ould Bouamama \(2008\)](#); [Venkatasubramanian \*et al.\* \(2003a,b,c\)](#), fault diagnosis methods can be classified in three general categories: quantitative model-based methods, qualitative model-based methods and process history based methods, as illustrated by Figure 2.2. The quantitative approach relies on advance information processing techniques such as state and parameter estimation and adaptive filtering. The qualitative approach makes use

of causal analysis (cause and effect/antecedents and consequences relationships), which links individual component malfunctions expressed in qualitative form with deviations in measured values. This approach can be used when precise numerical models are not available, especially in the design stage. Further sub-classifications and detailed reviews on each approach are given in the previous references.

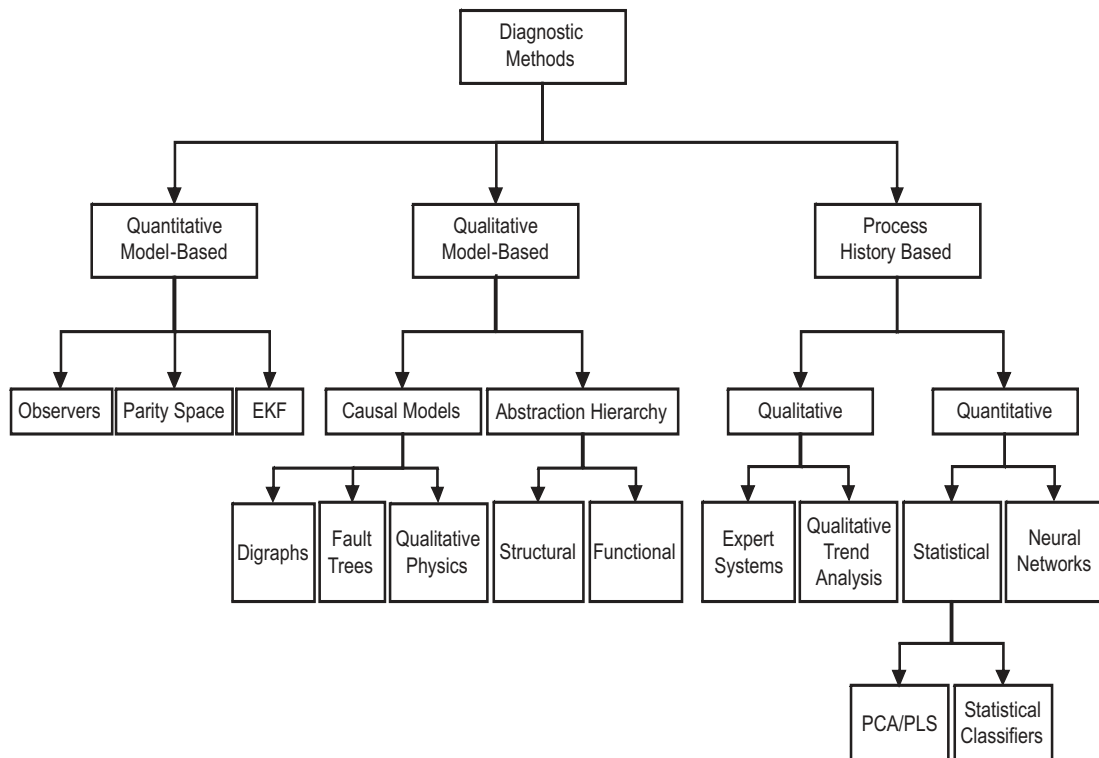


Figure 2.2: Classification of diagnosis methods.

The research that I have done in this PhD thesis includes several methods within process history based and model-based categories. Regarding process history based methods, firstly an application of an expert system to induction motors was developed (see [Acosta \*et al.\* \(2006\)](#)). Secondly, the results obtained by using statistical methods to locate the source of short duration faults (voltage sags) in a power distribution system, were the starting point for the work presented in [Khosravi \*et al.\* \(2007, 2008\)](#). This thesis is focused on model-based methods, which are reviewed in detail in the next section.

## 2.3 Research within Model-based Diagnosis

The so-called model-based diagnosis (MBD) approaches use an explicit model of the system to be diagnosed. Models are representations of knowledge about physical systems. The models used in MBD can be obtained from physical laws, human experience, data from the process, or from combination of the above.

A model for fault diagnosis should neither be too complex (i.e. include minor or secondary dynamics) nor be too simple (i.e. exclude some essential dynamics) [Samantaray & Ould Bouamama \(2008\)](#). Therefore, there needs to be some trade-off to create a reduced order process model, which would reliably replicate essential process dynamics under given operating conditions.

Since different levels of abstraction can be needed, properties such as precision, uncertainty or accuracy are reduced, increased or even lost. The properties of the models used in MBD have strong influence on the diagnosis results [Chantler \*et al.\* \(1998\)](#).

- Accuracy is the degree of closeness of a measured or calculated quantity to its actual (true) value.
- Precision is the degree to which further measurements or calculations show the same or similar results.

The applied process models can be classified according to the different descriptions of the variables with respect to time [Blanke \*et al.\* \(2006\)](#) :

- Continuous models. The evolution of continuous variables, whose values are in the set of real numbers, can be described in continuous or in discrete time. They are, in general, equation-based with further subclasses as linear, nonlinear, time-variant, time-invariant.
- Discrete-event models. The evolution of qualitative (or symbolic) variables is best described using discrete-event models such as automata, Petri nets, set of rules.
- Hybrid models. The time evolution of systems in which continuous variables and qualitative variables co-exist is described by so-called hybrid models.

Two research communities work on model-based techniques: the FDI (Fault Detection and Isolation) community, formed by researchers with a background in control systems engineering, and the DX (Principles of Diagnosis) community, formed by researchers with a background in computer science and intelligent systems. The collaboration between these two communities to develop more powerful tools for fault detection and diagnosis has been one of the goals of the European Network of Excellence on Model Based Systems and Qualitative Reasoning (MONET) [MONET \(1998\)](#), particularly of the Bridge task group. In the last years, several works have been developed on the “bridge” between DX and FDI, e.g. [de Kleer & Kurien \(2003\)](#) and in the Special Issue of the IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics, titled “Diagnosis of Complex Systems: Bridging the Methodologies of the FDI and DX Communities” (e.g. [Cordier \*et al.\* \(2004\)](#); [Gentil \*et al.\* \(2004\)](#); [Pulido & Alonso-González \(2004\)](#)).

This thesis is under the influence of the FDI and DX Communities, and because of this, the main concepts and approaches of both communities are outlined.

### 2.3.1 DX Community

In the late 1980s a group of Artificial Intelligence researchers independently proposed a fault diagnosis theory based on First-Order Logic. The system is modeled using the set of basic components of the system and the connections between them. The diagnosis consists in identifying the possible faulty components via an inference process. The papers laying the foundations of this theory are [de Kleer & Williams \(1987\)](#); [Reiter \(1987a\)](#). A more recent survey on this approach may be found, for example, in the collections [Hamscher \*et al.\* \(1992\)](#); [Struss \(2008\)](#).

DX community has mainly relied upon consistency-based techniques [Reiter \(1987b\)](#), but other approaches exist such as abductive methods [de Kleer \*et al.\* \(1992\)](#).

The consistency-based approach to diagnosis uses a model of a system to predict its expected behavior which can be compared with the actual behavior of the system (Fig. 2.3) [Dressler & Struss \(1996\)](#). If discrepancies are detected, there is a diagnosis problem. The device was correctly designed and built, but now

it exhibits indications of malfunctioning. The diagnosis task is to determine what is wrong in order to enable the re-establishment of the intended functionality.

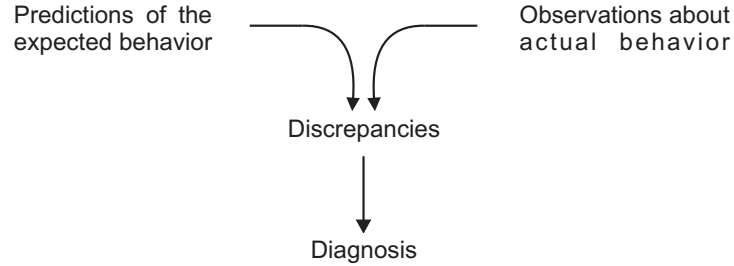


Figure 2.3: Basic Diagnosis Scheme.

**Definition 2.3.1.** *A system is a triple  $(SD, COMPS, OBS)$  where: (1)  $SD$ , is the system description; (2)  $COMPS$ , the system components, is a finite set of constants; (3)  $OBS$ , a set of observations, is a set of first-order sentences [Reiter \(1987b\)](#).*

A system is described by its system description, i.e. its model. A component can be in one of several different behavioral modes. Typically, each component  $c \in COMPS$  has an abnormal mode  $AB(c)$ , a normal mode  $\neg AB(c)$ , and one or several specific fault modes. It is sometimes preferable to only consider the  $AB$  and the  $\neg AB$  mode where the  $AB$  mode does not have a model [Biteus \(2007\)](#). In this way, only the normal behavior of a component has to be modeled, and the number of behavioral modes that has to be considered is reduced.

**Definition 2.3.2.** *A symptom is any difference between a prediction made by the inference procedure and an observation. Each symptom indicates that one or more components may be faulty.*

**Definition 2.3.3.** *A conflict is a set of assumptions which support a symptom, and thus leads to an inconsistency [de Kleer & Williams \(1987\)](#).*

**Definition 2.3.4.** *A candidate is represented by a set of assumptions. The assumptions explicitly mentioned are false, while the ones not mentioned are true. A candidate which explains the current set of symptoms is a set of assumptions such that if every assumption fails to hold, then every known symptom is explained. Thus each set representing a candidate must have a non-empty intersection with every conflict [de Kleer & Williams \(1987\)](#).*

## 2.3 Research within Model-based Diagnosis

---

The candidate space is the set of candidates consistent with the observations. The size of the initial candidate space grows exponentially with the number of components  $n$ . Any component could be working or faulty, thus the candidate space initially consists of  $2^n$  candidates [de Kleer & Williams \(1987\)](#).

**Definition 2.3.5.** *A diagnosis is a set of components  $\Delta \subseteq COMPS$  such that*

$$SD \cup OBS \cup \left\{ \bigwedge_{c \in \Delta} AB(c) \bigwedge_{c \in \Delta^c} \neg AB(c) \right\}$$

*is consistent* [de Kleer & Kurien \(2003\)](#).

The goal in single fault diagnosis is to obtain a unique single fault that can explain the observations. It may not always be possible to determine a unique candidate given the type of information used for diagnosis and the sensors available on the system [Daigle \(2008\)](#). In multiple fault diagnosis, the goal is expanded to obtain sets of faults that, taken together, explain the observations. Approaches for multiple fault diagnosis are more complex than the ones for single fault diagnosis. This is mainly due to a couple of reasons: (i) the effects of a fault could be masked or compensated by the effects of another fault, and (ii), the same multiple fault can manifest in different ways [Daigle et al. \(2006\)](#).

Within consistency-based diagnosis, the General Diagnosis Engine (GDE) is its most well-known implementation [de Kleer & Williams \(1987\)](#). GDE, as described in [Dressler & Struss \(1996\)](#), (i) computes candidates for diagnosis from minimal conflicts, i.e. minimal sets of component mode assumptions derived from detected inconsistencies, (ii) handles multiple faults, in contrast to previous systems, (iii) exploits an assumption-based truth maintenance system (ATMS [de Kleer \(1986\)](#)) to identify conflicting assumption sets, and (iv) uses it as a basis for determining optimal probing points. In GDE, four major phases are organized in an iterative cycle: (i) behavior prediction, (ii) conflict detection, (iii) diagnoses generation and ranking, and (iv) discrimination between diagnoses by additional measurements.

The main advantage of diagnosis systems based on consistency is that the ways in which a component fails are not needed, it is only needed a model of the correctly functioning device. However, one of the drawbacks of not including knowledge about possible faults, is that results may be logically possible, but



physically impossible. This situation can be seen in the example with three light bulbs described in [Dressler & Struss \(1996\)](#), where the diagnosis with fault modes is analyzed.

As it is stated in [Dressler & Struss \(1996\)](#), a consistency-based diagnosis algorithm based on models of correct behavior needs more probes to single out the final diagnosis. This is a severe limitation when not every point in a device is accessible for probing. Making use of knowledge about the behavior of faulty components can compensate for lacking observations. Hypothesizing a fault and then checking the consistency with what is known sometimes allows one to refute it. Of course, if no other diagnosis can be found an unknown fault must be present. If we assume, however, that we can enumerate all possible faults, a component can actually be exonerated.

Some of the most difficult issues to address in MBD within the DX community are discussed in [de Kleer & Kurien \(2003\)](#). They are: (i) toleration of noise in observable variables, (ii) diagnosis of hybrid discrete/continuous systems including, for example, continuous degradation in addition to discrete failure modes, and (iii), development or discovery of models adequate for diagnosis without excessive human engineering work. These issues are extensively studied in the FDI community.

### 2.3.2 FDI Community

This community has approached the diagnosis task from different perspectives as can be found in extensive surveys [Blanke \*et al.\* \(2006\)](#); [Chen & Patton \(1998\)](#); [Gertler \(1998\)](#); [Isermann \(2006\)](#); [Patton \*et al.\* \(2000\)](#); [Simani \*et al.\* \(2003\)](#).

A model-based fault diagnosis system normally consists of the following three tasks: detection, isolation and identification (Fig. 2.4). They are summarized as follows [Chen & Patton \(1998\)](#):

- Fault detection: to make a binary decision - either that something is wrong or that everything works under the normal conditions.
- Fault isolation: to determine the location of the fault, e.g. which component has become faulty.

## 2.3 Research within Model-based Diagnosis

---

- Fault identification: to estimate the size and type or nature of the fault.

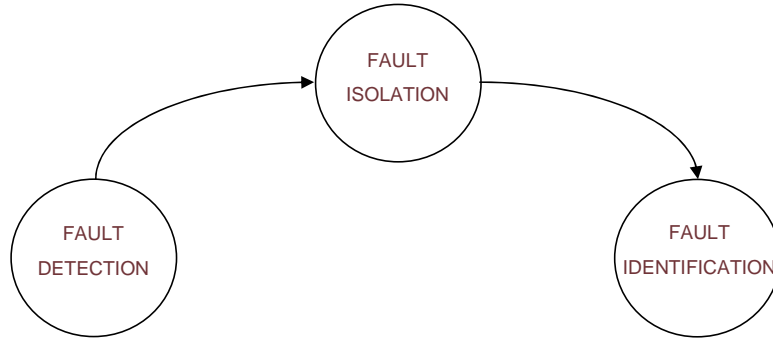


Figure 2.4: The three tasks of a model-based fault diagnosis system.

A fault-tolerant control system is a controlled system that continues to operate acceptably following faults in the system or in the controller [Palade \*et al.\* \(2006\)](#). An important feature of such a system is automatic reconfiguration, once a malfunction is detected and isolated.

Moreover, the fault diagnosis process can be viewed as a two-stage process, i.e. residual generation and decision making based on this residual [Chow & Willsky \(1980, 1984\)](#), and it has the structure shown in Fig. 2.5.

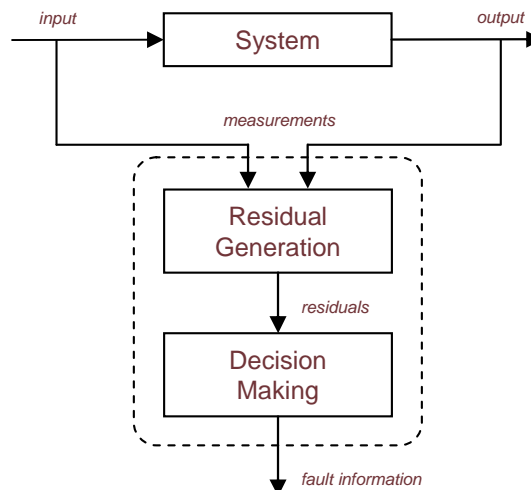


Figure 2.5: Two-stage structure of model-based FDI procedure.

- Residual Generation: its purpose is to generate a fault indicating signal - residual, using available input and output information from the monitored system. The resulting difference generated from the comparison of available system measurements with a priori information represented by the system's mathematical model is called a residual or symptom signal. The residual should be normally zero or close to zero when no fault is present, whilst distinguishably different from zero when a fault occurs. The algorithm used to generate residuals is called a residual generator [Chen & Patton \(1998\)](#).
- Decision-Making: this block examines residuals for the likelihood of faults and a decision rule is then applied to determine if any faults have occurred. The decision procedure may perform a simple threshold test on the instantaneous values or moving averages of the residuals. Moreover, it may consist of statistical methods, e.g., generalized likelihood ratio testing or sequential probability ratio testing [Basseville \(1988\)](#); [Chen & Patton \(1998\)](#); [Isermann \(2006\)](#). Most contributions in the field of quantitative model-based FDI focus on the residual generation problem.

Generally, the most popular approaches can be split into one of the following categories [Gertler \(1998\)](#): diagnostic observers, parity (consistency) relation, and parameter estimation.

- Observer-based approaches. The basic idea in behind the observer or filter-based approaches is to estimate the outputs of the system from the measurements (or a subset of measurements) by using either Luenberger observer(s) in a deterministic setting or Kalman filter(s) in a stochastic setting. Observers are dynamic systems that are aimed at reconstructing the states  $\mathbf{x}$  of a state-space model on the basis of the measured inputs  $\mathbf{u}$  and outputs  $\mathbf{y}$  [Pouliezos & Stavrakakis \(1994\)](#). Then, the (weighted) output estimation error (or innovations in the stochastic case), is used as a residual [Chen & Patton \(1998\)](#).
- Parity (consistency) relation. The parity space approach yields a systematic exploitation of the analytical redundancy provided by the mathematical

model of the system [Pouliezos & Stavrakakis \(1994\)](#). The basic idea in the parity relation approach is to provide a proper check of the parity (consistency) of the measurements of the monitored system [Chen & Patton \(1998\)](#). The parity relation approach and the observer-based approach have been widely compared, and they are equivalent in certain conditions, see for example [Christophe \*et al.\* \(2002\)](#); [Patton & Chen \(1991\)](#).

- **Parameter Estimation.** This approach is based on the assumption that the faults are reflected in the physical system parameters such as friction, mass, viscosity, resistance, inductance, capacitance, etc. The basic idea of the detection method is that the parameters of the actual process are repeatedly estimated on-line using well known parameters estimation methods and the results are compared with the parameters of the reference model obtained initially under the fault-free condition. Any substantial discrepancy indicates a fault [Chen & Patton \(1998\)](#).

## 2.4 Robustness in MBD

Uncertainty has been a common theme in several fields, including statistics, engineering, science and management. It is a term used to express that nothing is perfect in this world, at least in the sense that we perceive it.

Considering a system it is possible to identify several sources of uncertainty, even both internal and environmental sources. For instance, there is always a gap between the intrinsic properties (or conditions) and the explicit knowledge about materials and machines, for that reason perfectly identical results according to design are never produced. On the other hand, external interferences may enter the system and alter its performance and modify the output signal. It is well known that all components experience drifts related to their natural deterioration and the working environmental conditions. Other sources of uncertainty always present in a system are related to the human resources. In this way, one engineering challenge is to control under uncertain conditions the uniformity and consistency of the processes.

In addition, any measurement system consists of many components, including sensors. Thus, no matter how accurate the measurement is, it is only an approximation or estimate of the true value of the specific quantity subject to measurement. Never the measured values can be totally assured due to noise, errors in the analog-to-digital conversion, bias, drift, nonlinearities, inaccuracies due to calibration, etc. The result of a measurement should be considered complete only when accompanied with a quantitative statement of its uncertainty.

Model based diagnosis makes use of mathematical models of the physical system, but a perfectly accurate and complete model is never available. Usually the parameters of the system may vary with time in an uncertain manner, and the characteristics of the disturbances and noise are unknown so that they cannot be modeled accurately. Hence, there is always a mismatch between the actual process and its mathematical model even under no fault conditions. Such discrepancies constitute a source of false and missed alarms [Chen & Patton \(1998\)](#); [Simani et al. \(2003\)](#). So there is a need to develop robust fault diagnosis algorithms. The robustness of a fault diagnosis system means that it must be only sensitive to faults, even in the presence of model-reality differences [Chen & Patton \(1998\)](#).

**Definition 2.4.1.** *The robustness problem in FDI is defined as the maximization of the detectability and isolability of faults together with the minimization of the effect of the uncertainty and disturbance on the FDI procedure [Patton \(1997\)](#).*

Two main families of approaches have been proposed to solve the robustness problem in the FDI community [Chen & Patton \(1998\)](#). One, based on an attempt to account for the uncertainty in designing the residual, is known as *active robustness* in FDI. It includes different techniques as for example, unknown input observers, robust parity equations, frequency domain design techniques (e.g.  $H_\infty$ -norm optimization), etc. The second family of approaches is called *passive robustness* in FDI and consist in enhancing the robustness of the fault detection procedure at the decision-making stage. In this latter case, robustness can be achieved by finding and using the most effective threshold.

The robustness problem has also been treated in the DX community, and appears in several books and papers of the artificial intelligence conferences and journals.

As it is described at the end of this section, the approaches can be put into three categories depending on the nature of the information: quantitative, qualitative, and semiquantitative.

### 2.4.1 Bounding Approaches

When the uncertainties are structured, i.e. the model structure is known and only the parameters undergo imprecisions, they can be handled with interval models in which the equation parameter values are allowed to vary within numeric intervals [Armengol \*et al.\* \(2000\)](#). By using interval models for fault detection, the actual behavior of the system is compared to a set of expected behaviors obtained by varying the uncertain parameters within their corresponding intervals. These approaches, known as set membership or unknown-but-bounded, are an alternative to statistical ones assuming a probabilistic description of uncertainty. They can be applied when, for example, there is a shortage of probabilistic information (e.g. it is not known the probability density function, pdf, that characterizes the uncertainty) because nothing is said about the behavior of the uncertain parameter or variable between the bounds [Milanese \*et al.\* \(1996\)](#); [Norton \(1994\)](#).

In many practical cases a bounded error assumption is more realistic and less demanding than statistical assumptions, or even more when the very random nature of uncertainty may be questionable [Milanese & Vicino \(1991\)](#); [Milanese \*et al.\* \(1996\)](#). For example, the real process generating the actual data may be very complex (large scale, nonlinear, time-varying) so that only simplified models can be practically used. Then, the residuals of the estimated model have a component due to deterministic structural errors, and treating them as purely random variables may lead to unsatisfactory results.

### 2.4.2 Robustness by using Quantitative, Qualitative, and Semiquantitative Information

For fault detection, the prediction of the normal behavior using interval models can be performed by several means, including quantitative, qualitative, and semiquantitative or semiquantitative methods. [Armengol \*et al.\* \(2000\)](#) surveys the state of the art of interval model simulators (considering a simulation as a prediction

across time) and their properties related to fault detection. The main properties include soundness, completeness, and stability. An estimate is complete if it includes all possible behaviors. On the other hand, if the estimate is too wide, it includes zones that cannot be reached by any of the quantitative models belonging to the set. Hence, the estimate is not sound. An overbounded estimate is complete but not sound, and this can result in missed alarms. A sound but not complete estimate is an underbounded estimate and may result in false alarms. The exact estimate is complete and sound but, in general, a way to obtain it is to solve a global optimization problem Hansen (1992), which usually requires considerable computational effort. Finally, the stability of an estimate is a desirable property related to the growth of the estimate width with time. If the estimate width grows with time, as the input signal remains the same, it is unstable.

- Quantitative simulation. It makes numeric predictions of the system states. The simulation of the behavior of interval models can be achieved (i) by using superimposed thresholds (constant or variable/adaptive), (ii) by selecting randomly (Monte Carlo) or systematically a number of quantitative models belonging to the set and then extracting conclusions for the whole set, and (iii) finding the maximum and the minimum of a function into a parameter space, i.e. solving a global optimization problem.
- Qualitative simulation. It makes a prediction of the qualitative states in which the system will be by using qualitative information about the relations between the variables. The qualitative simulation can be classified into two types: non constructive and constructive. Non constructive qualitative simulation consists of two phases Coghill (1996): TG (Transition Generation), in which the transitions are generated, and QA (Qualitative Analysis), in which the states are filtered using the model to eliminate the ones that do not fulfil the constraints (constraint propagation). Most of the qualitative simulators of this type are not causal and do not hence consider time explicitly. Some qualitative simulators are, for example, QSIM (Qualitative Simulator) Kuipers (1990, 1994), PA (Predictive Algorithm) Wiegand (1991).

- Semiquantitative simulation. Some numerical knowledge is available, as in interval models. In this manner, qualitative methods can be enhanced, as for example in Q2 Kuipers & Berleant (1988), Q3 Berleant & Kuipers (1992), SQSIM Kay (1996). Moreover, some semiquantitative simulators are based on fuzzy sets, or interval arithmetic, e.g. Ca En (Causal Engine) Bousson & Travé-Massuyès (1994); Travé-Massuyès *et al.* (2001), Simulator of Gasca *et al.* (2002, 1996).

### 2.4.3 Interval Techniques used for Fault Detection

Over the last years, many works using interval techniques for fault detection have appeared.

These techniques mainly include the ones based on:

- Classical interval analysis, e.g. in Fagarasan *et al.* (2004); Karim *et al.* (2008).
- Modal interval analysis, e.g. in Armengol *et al.* (2003, 2009b).
- Set membership identification methods, e.g. in Combastel *et al.* (2008); Ingimundarson *et al.* (2005, 2009); Janati-Idrissi *et al.* (2002); Watkins & Yurkovich (1996).
- Constraint propagation techniques, e.g. in Gelso *et al.* (2007c); Stancu *et al.* (2003).

A more in depth description of this kind of fault detection techniques is presented in the related work of Chapter 4.





# Chapter 3

## Residual Generation

### Summary

Structural models can provide much useful information for fault diagnosis and fault-tolerant control design, since structural analysis is able to identify those components of the systems which are, or are not, monitorable, to provide design approaches for analytical redundancy based residuals, and to identify those components whose failure can, or cannot, be tolerated through reconfiguration [Blanke \*et al.\* \(2006\)](#).

In this chapter, a residual generation toolbox based on structural analysis and developed in Matlab<sup>®</sup> is proposed. The toolbox includes algorithms to find automatically all minimal structurally overdetermined (MSO) sets in a structural model of a system, all causal models for each MSO set, and routines to represent them and to analyze the detectability and isolability of faults. Moreover, it includes algorithms for investigation of the possibilities of active structural isolation to enhance the structural isolability of faults.

The work presented in this chapter was partially developed during my research stays in Denmark, under the supervision of Prof. Mogens Blanke, and in Valladolid, under the supervision of Prof. Belarmino Pulido.

## 3.1 Related Work

The task of designing efficient diagnosis procedures becomes more difficult when the complexity of the system increases because of, for example, the increase of the number of components. Therefore, an important task to develop is the design of detection and isolation tests in order to overcome faults quickly by obtaining a fast and correct diagnostic. In the context of this thesis, an abstraction of the behavior model of a system, the structural model, is used to obtain the complete set of overconstrained, or overdetermined, subsystems, and thus, to achieve an optimal diagnostic.

These fault tests can take the form of parity relations or analytical redundancy relations (ARR) [Blanke \*et al.\* \(2006\)](#), and can be deduced by means of Structural Analysis (Section 3.2). Once they are designed (off-line), the fault detection procedure checks on-line the consistency of the observations with respect to every of these tests. When discrepancies occur between the modeled behavior and the observations (non-zero residual signals), the fault isolation procedure identifies the system component(s) which is (are) suspected of causing the fault [Cordier \*et al.\* \(2004\)](#).

Being the obtainment of minimal structurally overdetermined (MSO) sets [Krysander \*et al.\* \(2008\)](#) a really complex task, some authors have developed approaches to deal with this issue.

A Matlab<sup>®</sup> toolbox called SaTool is presented in [Blanke & Lorentzen \(2006\)](#). It is an implementation of the structural analysis theory [Staroswiecki \(2002\)](#) and extends the work of Staroswiecki and co-workers. A ranking algorithm [Blanke \*et al.\* \(2006\)](#) is used to find a complete matching in the structure graph. The overdetermined sets are obtained by means of the constraints that are not involved in the complete matching. This version of the toolbox, which does not deliver all the MSO sets, is the starting point of the algorithm proposed in this chapter.

In [Izadi-Zamanabadi \(2002\)](#), as contrasted with the algorithm cited above, an algorithm based on determining several matchings is presented. Different matchings result in different overdetermined subsystems. The definition presented in this paper of a minimal overdetermined subsystem is equivalent to the definition of a MSO set.

Furthermore, in [Düşttegör \(2005\)](#), an algorithm generates the set of all complete matchings by performing a depth-first-traversal of the bipartite graph. By this way, it is possible to gather all the possible residuals structure, without loss of information, in order to be able to distinguish different faults from each other. In the worst-case, for a fixed order of structural redundancy, the computational complexity of this algorithm is factorial in the number of equations. Because of this reason, this algorithm can be very time consuming, or even worse, for large systems it can be impossible to apply. One improvement is also proposed in [Düşttegör \(2005\)](#) to decrease the number of matchings to consider. This improvement is based on the connectedness property of the König-Hall blocks.

In [Krysanter \*et al.\* \(2008\)](#), an efficient algorithm based on a top-down approach is presented. It starts with the entire model and then reduces the size of it step by step until an MSO set remains.

Each constraint of a MSO set can be interpreted in a causal way (in the sense of each variable may be deduced from the others), leading to different causal interpretations, see Section 3.5. In this sense, information about causality can be taken into account to discard unachievable overdetermined sets, as it is done in e.g. [Düşttegör \*et al.\* \(2004\)](#); [Ploix \*et al.\* \(2005\)](#); [Pulido & Alonso-González \(2004\)](#); [Travé-Massuyès \*et al.\* \(2006\)](#).

In [Ploix \*et al.\* \(2005\)](#), an algorithm based on elimination rules is presented. It combines constraints in order to eliminate all the unknown variables and therefore getting constraint containing only known data.

A variant of the method presented in [Travé-Massuyès \*et al.\* \(2006\)](#), which was devised for diagnosability assessment, is used to obtain the MSO sets which have at least one causal interpretation. Starting from a structural model augmented by causal information, the algorithm successively eliminates unknown variables on the structural model equations.

[Pulido & Alonso-González \(2004\)](#) proposes the Possible Conflict Computation (PCC) algorithm. The algorithm first determines MSO sets, and then searches for all the causally consistent interpretations for each of them. For determining MSOs, it performs successive unknown variable eliminations in a depth-first search manner. Causal interpretations are propagated in the same way for each MSO set.

Düştögör *et al.* (2004) can also deal with causality. To find a matching, a method based on a class of algorithms that solve the Stable Marriage Problem is presented and adapted for the fault detection purpose. In mathematics, the Stable Marriage Problem is a well-known combinatorics problem dealing with finding a stable matching, i.e. a matching in which no element of the first matched set prefers an element of the second matched set that has not the inverse preference.

Verde *et al.* (2007) discusses the diverse complete matchings that can be used to generate the residual structure based on input-to-output causal graphs. It is shown that the number of matchings is reduced by incorporating the knowledge of cause and effect variables, i.e. the known exogenous and endogenous variables.

For non-linear polynomial models, algorithms based on elimination techniques, e.g. Gröbner Basis, can be used to obtain analytical redundancy relations of the system Ceballos *et al.* (2004); Frisk (2000); Staroswiecki & Comtet-Varga (2001). As basis of this method, model equations are manipulated to eliminate unknown variables such as disturbances, faults and internal states.

## 3.2 Structural Analysis

In the structural analysis Staroswiecki (2002), the structure of a model  $M$  (or the structural model) of the system  $(M, X \cup Z)$  is represented by a bipartite graph  $G(M, X \cup Z)$  (or equivalently its biadjacency or incidence matrix) with variables and constraints as node sets.  $X$  is the set of unknown variables and  $Z$  is the set of known variables. There is an edge  $(c_i, v_j) \in \mathcal{E}$  (set of edges) connecting a constraint  $c_i \in M$  and a variable  $v_j \in X \cup Z$  if  $v_j$  is included in  $c_i$  Blanke *et al.* (2006).

The incidence matrix for a bipartite graph  $G$  is the  $n \times m$  matrix  $S$ , where  $n$  is the number of constraints and  $m$  is the number of variables, such that  $S(i, j) = 1$  if the variable  $v_j$  appears in constraint  $c_i$ .

**Example 3.2.1.** *Let a system be given by the structure graph shown in Fig. 3.1. It is described by seven constraints  $\{c_1, c_2, c_3, c_4, c_5, c_6, c_7\}$ , and eleven variables:  $\{u_1, u_2\}$  are the control variables,  $\{y_1, y_2, y_3, y_4\}$  are the sensor outputs, and  $\{x_1, x_2, x_3, x_4, x_5\}$  are the unknown variables. The associated incidence matrix is shown in Table 3.1, and the bi-partite graph is depicted in Fig. 3.2.*

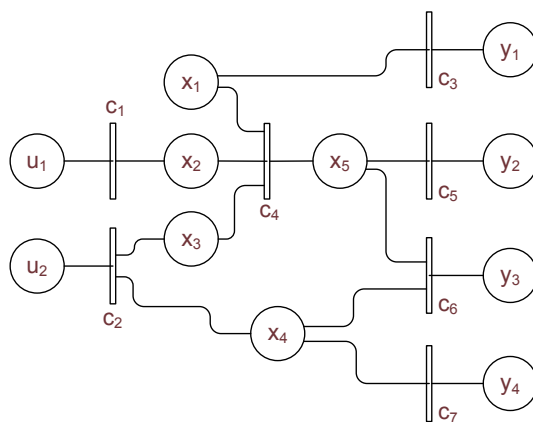


Figure 3.1: Structure graph of Example 3.2.1.

Table 3.1: Incidence matrix of Example 3.2.1.

	$u_1$	$u_2$	$y_1$	$y_2$	$y_3$	$y_4$	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$
$c_1$	1	0	0	0	0	0	0	1	0	0	0
$c_2$	0	1	0	0	0	0	0	0	1	1	0
$c_3$	0	0	1	0	0	0	1	0	0	0	0
$c_4$	0	0	0	0	0	0	1	1	1	0	1
$c_5$	0	0	0	1	0	0	0	0	0	0	1
$c_6$	0	0	0	0	1	0	0	0	0	1	1
$c_7$	0	0	0	0	0	1	0	0	0	1	0

When considering differential algebraic systems, different alternatives for handling derivatives exist [Düşttegör \*et al.\* \(2006\)](#); [Krysander \*et al.\* \(2008\)](#).

- There is no distinction between one variable and its derivative.
- Unknowns and their time derivatives are, in contrast to previous representation, considered to be separate independent algebraic variables. New equations can be obtained by differentiation.
- Unknowns and their time derivatives are, as in the second representation, considered to be separate independent algebraic variables. Thus, the equations are purely algebraic, and differential relations that link one variable and its derivative are added.

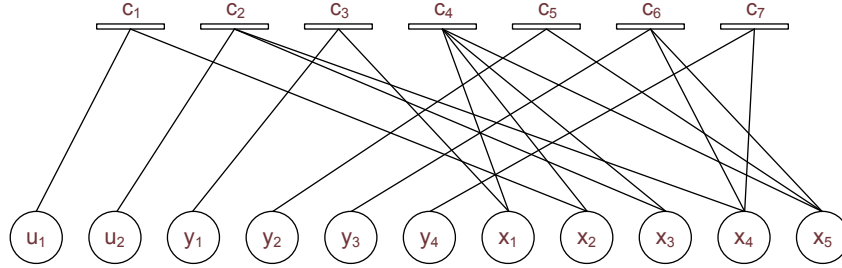


Figure 3.2: Bi-partite graph of Example 3.2.1.

In this work, in spite of all the representations can be applied to the proposed algorithms, the third alternative is preferred. As explained later, the differential relations that link one variable and its derivative can be used, for example, to establish a derivative causality or an integral causality.

Considering the following definitions:

**Definition 3.2.1.** *Matching.* A matching  $\mathcal{M}$  is a subset of edges  $\mathcal{E}$  such that with  $e_1 = (c_{i1}, v_{j1})$ ,  $e_2 = (c_{i2}, v_{j2})$ ,  $\forall e_1, e_2 \in \mathcal{M} : e_1 \neq e_2 \Rightarrow c_{i1} \neq c_{i2} \wedge v_{j1} \neq v_{j2}$ .

**Definition 3.2.2.** *Complete matching on  $X$ .* A matching is complete on  $X$  if  $|\mathcal{M}| = |X|$ .

A matching is a causal assignment which associates unknown system variables with the system constraints from which they can be calculated [Blanke et al. \(2006\)](#). If it is possible to match all unknown variables, the matching will identify over-determined subgraphs (those which contain more equations than unknown variables) that can be used to obtain analytical redundancy relations in the system. In particular, the minimal over-determined subgraphs are useful for the fault detection and isolation task.

**Definition 3.2.3.** A set  $M$  of equations is structurally overdetermined (SO) if  $M$  has more equations than unknowns [Krysander et al. \(2008\)](#).

**Definition 3.2.4.** An SO set is a minimal structurally overdetermined (MSO) set if no proper subset is an SO set [Krysander et al. \(2008\)](#).

Using the MSO sets and causal information about calculability of the constraints of the model, parity relations (or Analytical Redundancy Relations, ARR) can be obtained. Analytical redundancy relations are static or dynamic

constraints which link only known variables when the system operates according to its normal operation model [Staroswiecki \(2002\)](#).

In this thesis, a residual generation toolbox called SRGlib (Structural based Residual Generation), which is based on structural analysis and developed in Matlab<sup>®</sup>, is proposed. This approach is organized in seven steps .

- Find a complete matching.
- Obtain a first set of MSO sets (basic MSO sets).
- “Combine” basic MSO sets to obtain the complete set.
- Find (all) causal models for each MSO set.
- Represent results as symbolic expressions and as oriented graphs.
- Perform the analysis of detectability and isolability of faults.
- Perform the analysis of active structural isolability of faults.

All of them are explained in the following sections.

### 3.3 Matching Algorithms

Several algorithms exist to find complete matchings on unknown variables in a structure graph, see for example [Bondy & Murty \(1976\)](#); [Diestel \(2000\)](#).

The ranking algorithm (Algorithm 1) can be used to achieve a complete matching on the unknown variables [Blanke \*et al.\* \(2006\)](#). The input of this algorithm is the incidence matrix  $S$  of  $G(M, X \cup Z)$ .

In the first step of Algorithm 1, all known variables are marked and all unknown variables remain unmarked. Then every constraint that contains at most one unmarked variable is assigned rank 1. The constraint is matched for the unmarked variables, and the variable is marked. This step is repeated with an increasing rank number, until no new variables can be matched. Every matched variable has associated a number, or rank, which can be interpreted as the number of steps needed to calculate an unknown variable from the known ones.



---

**Algorithm 1** Ranking Algorithm

---

**Input:** Incidence matrix

- 1: Mark all known variables,  $i=1$ ;
  - 2: Find all constraints with exactly one unmarked variable.  
Associate rank  $i$  with these constraints.  
Mark these constraints and the associated variable;
  - 3: If there are unmarked constraints whose variables are all marked, mark them and connect them with the pseudovariable zero;
  - 4: set  $i = i+1$ ;
  - 5: if there are unmarked variables or constraints, continue with step 2;
  - 6: **return** Ranking of the constraints
- 

The ranking algorithm does only generate oriented graphs without loops, so it may not find any complete matching, even if it exists [Blanke \*et al.\* \(2006\)](#).

Algorithm 2 is proposed in this thesis to find one or all complete matchings on unknown variables, in a structure graph with or without loops. In contrast to the ranking algorithm, the proposed algorithm finds, in the second step, all variables with exactly one unmarked constraint. In the seventh step, two algorithms can be called to match unmarked variables appearing in at least two constraints. On the one hand, Algorithm 3 finds only one matching of unmarked variables. It is a recursive algorithm with a stop criterion when all unmarked variables are matched. It can use a heuristic to rapidly come to a complete matching, if it exists. For example in step 4, the criterion of choosing a constraint can be the ranking of the constraints calculated by the ranking algorithm. On the other hand, Algorithm 4 finds all possible matchings of unmarked variables, and in this way, it is used to find all complete matchings on unknown variables of the incidence matrix. It exhaustively searches the entire graph in a non-recursive way.

---

**Algorithm 2** Proposed Matching Algorithm
 

---

**Input:** Incidence matrix

- 1: Mark all known variables,  $i=1$ ;
  - 2: Find all variables appearing exactly in one unmarked constraint.  
Associate rank  $i$  with these variables and constraints.  
Mark these variables and the corresponding constraints;  
Add them to the matching;
  - 3: If there are unmarked variables whose constraints are all marked, stop: a complete matching can not be found;
  - 4: set  $i = i+1$ ;
  - 5: **if** there are unmarked variables, **then**
  - 6:   **if** no variables were found in step 2, **then**
  - 7:     run Algorithm 3 or 4 for the unmarked variables and constraints;  
   associate rank  $i$  with these variables.
  - 8:   **else**
  - 9:     continue with step 2;
  - 10:   **end if**
  - 11: **end if**
  - 12: **return** Matching and Ranking of the constraints
- 

---

**Algorithm 3** one\_matching function
 

---

$(matching, end\_algorithm) = \text{find\_one\_matching}(unmarked\_var, unmarked\_con)$

- 1:  $matching := \emptyset$ ;  $end\_algorithm := 0$
  - 2: choose a variable  $v$ ,  $v \in unmarked\_var$ ;
  - 3:  $unmarked\_var = unmarked\_var \setminus \{v\}$ ;
  - 4: **for all** constraint  $c \in unmarked\_con$  such that  $c$  contains  $v$  **do**
  - 5:   **if**  $unmarked\_var = \emptyset$  **then**
  - 6:      $matching := (v, c)$ ;
  - 7:      $end\_algorithm := 1$ ;
  - 8:     break;
  - 9:   **else**
  - 10:      $(matching, end\_algorithm) := \text{find\_one\_matching}(unmarked\_var,$   
 $unmarked\_con \setminus \{c\}$ );
  - 11:   **if**  $end\_algorithm = 1$  **then**
  - 12:      $matching := matching \cup (v, c)$ ;
  - 13:     break;
  - 14:   **else**
  - 15:      $matching := \emptyset$ ;
  - 16:   **end if**
  - 17: **end for**
  - 18: **end for**
  - 19: **return**  $matching, end\_algorithm$
-

**Algorithm 4** all\_matchings function

$matchings = \text{find\_all\_matchings}(unmarked\_var, unmarked\_con)$

---

```

1:  $matchings := \emptyset$ ;
2:  $i := 0$ ;
3: for all variable  $v$  in  $unmarked\_var$  do
4:    $i := i + 1$ ;
5:   mark  $v$ ;
6:   if  $i = 1$  then
7:     for all constraint  $c \in unmarked\_con$  such that  $c$  contains  $v$  do
8:        $matchings := matchings \cup \{(v, c)\}$ ;
9:     end for
10:  else
11:     $matchings\_temp := \emptyset$ ;
12:    for all matching  $m$  in  $matchings$  do
13:      for all constraint  $c \in unmarked\_con$  such that  $c$  contains  $v$  do
14:        if  $c$  is not matched in  $m$  then
15:           $m := m \cup (v, c)$ ;
16:           $matchings\_temp := matchings\_temp \cup m$ ;
17:        end if
18:      end for
19:    end for
20:    if  $matchings\_temp \neq \emptyset$  then
21:       $matchings := matchings\_temp$ ;
22:    else
23:      break; return no matching;
24:    end if
25:  end if
26: end for
27: return  $matchings$ 

```

---

In this work, Algorithms 2 and 3 are used together to find one complete matching, and in this way, basic MSO sets. Algorithms 2 and 4 are used together to find all causal models for each MSO set (see Section 3.5).

**Example 3.3.1.** Consider again the Example 3.2.1, in this system, one complete matching on unknown variables is obtained using Algorithms 2 and 3. The edges of the complete matching are identified by a “Ⓛ” in the incidence matrix of Table 7.5, and by thick lines in the bi-partite graph illustrated in Fig. 3.3.

Table 3.2: Incidence matrix with unknown variables showing one complete matching.

	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	Ranking
$c_1$	0	Ⓛ	0	0	0	1
$c_2$	0	0	Ⓛ	1	0	2
$c_3$	Ⓛ	0	0	0	0	1
$c_4$	1	1	1	0	1	2
$c_5$	0	0	0	0	Ⓛ	1
$c_6$	0	0	0	1	1	2
$c_7$	0	0	0	Ⓛ	0	1

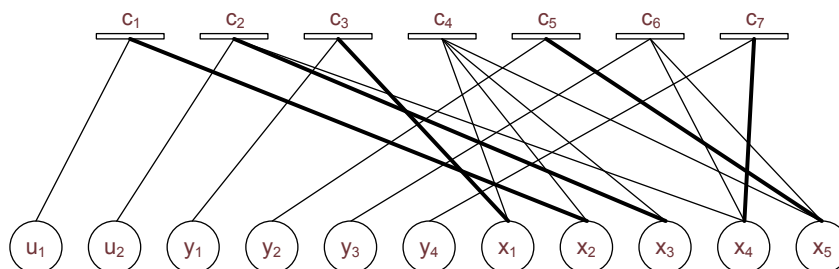


Figure 3.3: A complete matching represented on a bi-partite graph.

Figure 3.4 shows the order in which the matched edges are expanded. Starting at variable  $x_1$ , it can be matched with  $c_3$  or  $c_4$ . After choosing  $c_3$ , now  $x_2$  can be matched with  $c_1$  or  $c_4$ . Iterative deepening the tree allows to obtain the matching showed with red thick lines. Sixteen complete matchings can be found by applying algorithms 2 and 4 to this example.

### 3.4 A Residual Generation Algorithm

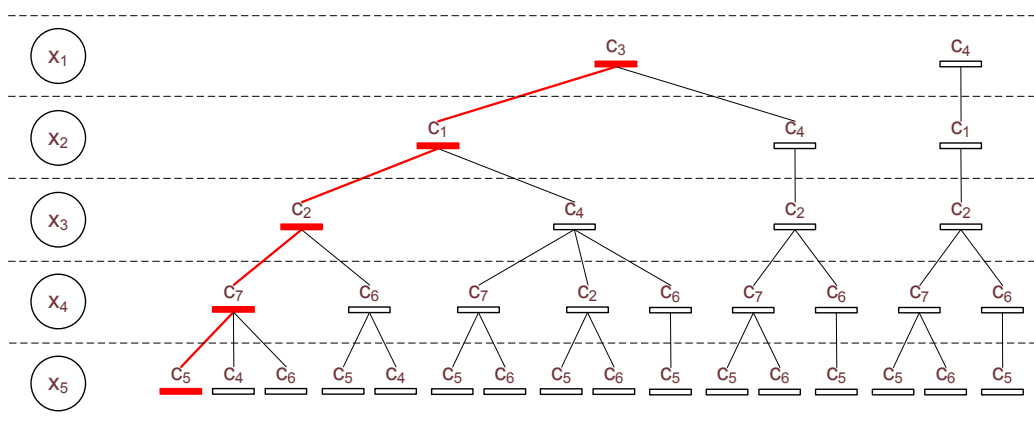


Figure 3.4: Order in which the matched edges are expanded.

## 3.4 A Residual Generation Algorithm

In this section, a new algorithm is presented for finding all MSO sets. Finding the complete set of MSO sets is useful for the diagnosis task to increase the fault isolability due to the fact that it can provide different signatures to each fault. The proposed algorithm is based on the fact that the basic MSO sets can be combined in order to get more MSO sets, using the information from one complete matching. Other algorithms like the ones in e.g. [Düştögör \(2005\)](#); [Laursen \*et al.\* \(2008\)](#), are based on finding all complete matchings, which is a difficult task because the computational complexity of these kind of algorithms is factorial in the number of constraints.

Unmatched constraints of one complete matching are used to obtain a first family of MSO sets, which will be called Basic MSO sets. Each MSO set is obtained from an unmatched constraint  $c_i$  by (i) backtracking unknown variables of  $c_i$  through constraints to which they were matched, or (ii), by the over-constrained subsystem of the canonical decomposition of the system which is composed of the set of the equations of the matching and  $c_i$ .

The canonical decomposition of bipartite graphs, discovered by Dulmage and Mendelsohn, states that any finite-dimensional graph can be decomposed into three subgraphs with specific properties, respectively associated with an over-constrained, a just-constrained, and an under-constrained subsystem. For more

### 3.4 A Residual Generation Algorithm

---

information about the canonical decomposition see, for example, [Dulmage & Mendelsohn \(1958, 1963\)](#).

Algorithm 5 is based on combining the basic MSO sets (the collection of basic MSO sets is called  $\mathcal{C}_{MSO_1}$  in the algorithm). A structurally overdetermined (SO) set can be obtained from the elimination of at least one shared equation from the set of equations of two MSO sets, and this operation is called a combination. Each SO set obtained by the combination of two MSO sets can be minimal, i.e. a MSO set, or not. This algorithm finds only the minimal ones.

Function ‘Combine’, which is presented as Algorithm 6, leads to obtain the new MSO sets after combining two collections of MSO sets. Step five in the algorithm is very important. It can be tackled in a brute-force way, which can result in a combinatorial explosion. This method avoids this by removing one shared constraint at a time, and the corresponding matched constraints used only to calculate the unknown variables of the shared constraint.

---

**Algorithm 5** Algorithm to find all MSO sets

---

**Input:** Complete matching

- 1:  $\mathcal{C}_{MSO_1} \leftarrow$  Basic MSO sets;
  - 2:  $i = 1$ ;
  - 3: **while**  $i <$  number of MSO sets in  $\mathcal{C}_{MSO_1}$ , or,  $\mathcal{C}_{MSO_i}$  is not empty **do**
  - 4:    $\mathcal{C}_{MSO_{i+1}} :=$  Combine( $\mathcal{C}_{MSO_i}, \mathcal{C}_{MSO_1}$ );
  - 5:   set  $i=i+1$ ;
  - 6: **end while**
  - 7:  $\mathcal{C}_{MSO} := (\mathcal{C}_{MSO_1} \dots \mathcal{C}_{MSO_i})$ ;
  - 8: **return** Complete set of MSO sets:  $\mathcal{C}_{MSO}$
-

### 3.4 A Residual Generation Algorithm

**Algorithm 6** Combine function

- 1:  $\mathcal{C}_{MSO_{ab}} = \text{Combine}(\mathcal{C}_{MSO_a}, \mathcal{C}_{MSO_b})$
- 2: **for all** set  $MSO_a$  in  $\mathcal{C}_{MSO_a}$  **do**
- 3:   **for all** set  $MSO_b$  in  $\mathcal{C}_{MSO_b}$  **do**
- 4:     **if** shared constraints set of  $MSO_a$  and  $MSO_b$  is not void, and,  $MSO_a$  and  $MSO_b$  do not share the same unmatched constraint **then**
- 5:       Remove alternately one or more shared constraints from  $MSO_a \cup MSO_b$ ;
- 6:       Check if each obtained SO set is minimal;
- 7:       If it is minimal, add to  $\mathcal{C}_{MSO_{ab}}$ ;
- 8:     **end if**
- 9:   **end for**
- 10: **end for**
- 11: **return**  $\mathcal{C}_{MSO_{ab}}$

**Example 3.4.1.** Consider the following incidence matrix:

	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$	<i>Ranking</i>
$c_1$	①							1
$c_2$	1	①						2
$c_3$		1	①					3
$c_4$				①	1			2
$c_5$					①			1
$c_6$						①	1	2
$c_7$							①	1
$c_8$			1	1				3
$c_9$			1			1		3

Matched constraints are  $c_1$  to  $c_7$ . Therefore two basic MSO sets can be found using  $c_8$  and  $c_9$ . The first one is  $\{c_1, c_2, c_3, c_4, c_5, c_8\}$ , and the second one is  $\{c_1, c_2, c_3, c_6, c_7, c_9\}$ . The shared constraints in both are  $\{c_1, c_2, c_3\}$ , and then, seven different possibilities of removing them are possible. Instead of this, using the information of the matched variables, we can see that  $c_3$  (rank 3) was matched using  $c_2$  (rank 2), and  $c_2$  using  $c_1$  (rank 1). Removing  $c_3$ , in this example also implies that  $c_1$  and  $c_2$  can be removed because they are not used for another calculation. So the third MSO set is  $\{c_4, c_5, c_6, c_7, c_8, c_9\}$ .

### 3.4.1 Computational Complexity

The computational complexity can be studied in two parts:

(i) for the matching algorithm used to find the basic MSO sets. For example, the ranking algorithm has complexity  $O(nm)$  where  $n$  is the number of constraints and  $m$  is the number of unknown variables [Blanke & Lorentzen \(2006\)](#).

(ii) for the algorithm used to combine the basic MSO sets. Being  $\varphi$  the structural redundancy of a model, the  $\varphi$  basic MSO sets are combined, using [Algorithm 5](#), in groups of 2 to  $\varphi$  (in the worst case). For each combination, there are at most  $r$  shared constraints to be removed.  $r$  is less than or equal to  $m$ . For a worst case, the function of operations can be expressed by the following function

$$\sum_{k=2}^{\varphi} \binom{\varphi}{k} r^{k-1} \leq \sum_{k=2}^{\varphi} \binom{\varphi}{k} m^{k-1} < \sum_{k=2}^{\varphi} \binom{\varphi}{k} m^{\varphi-1} = m^{\varphi-1}(2^{\varphi} - \varphi - 1) \quad (3.1)$$

Hence, for a fixed order of structural redundancy, the computational complexity of the algorithm is polynomial in the number of unknowns. This condition makes this algorithm suitable for real-world systems with a large number of unknown variables and constraints, but low structural redundancy (which depends on the number of available sensors) [Krysander \*et al.\* \(2008\)](#).

Proof:

$\varphi = n - m$  is the structural redundancy of a model. Then there are  $\varphi$  basic MSO sets.

The basic MSOs are combined in groups of 2 ( $k = 2$ ), there are  $\binom{\varphi}{2} = \frac{\varphi(\varphi-1)}{2}$  groups. For each combinations there are at most  $r$  shared constraints to be removed.  $r$  is less than or equal to  $m$ .

$$\binom{\varphi}{2} r \leq \binom{\varphi}{2} m \quad (3.2)$$

For  $k = 3$ , in the worst case, the previous sets are combined with the remaining  $\frac{\varphi-2}{3}$  basic MSO sets (without repetitions), and the shared constraints are at most



$m - 1$ .

$$\left[ \binom{\varphi}{2} m \right] \frac{\varphi - 2}{3} (m - 1) = \binom{\varphi}{3} m(m - 1) < \binom{\varphi}{3} m^2 \quad (3.3)$$

And the same procedure for  $k = 4$  and so on.

$$\left[ \binom{\varphi}{3} m(m - 1) \right] \frac{\varphi - 3}{4} (m - 2) = \binom{\varphi}{4} m(m - 1)(m - 2) < \binom{\varphi}{4} m^3 \quad (3.4)$$

The general formula is

$$\begin{aligned} \sum_{k=2}^{\varphi} \binom{\varphi}{k} (m)_{k-1} &< \sum_{k=2}^{\varphi} \binom{\varphi}{k} m^{k-1} < \\ &< \sum_{k=2}^{\varphi} \binom{\varphi}{k} m^{\varphi-1} = m^{\varphi-1} (2^{\varphi} - \varphi - 1) \end{aligned} \quad (3.5)$$

where  $(m)_{k-1}$  is the permutations without repetitions of  $m$  in groups of  $k - 1$ , and it is equal to  $\frac{m!}{(m-(k-1))!}$ .

In the previous expression, the second term can be expressed as the expansion of the binomial  $(x + y)^z = \sum_{k=0}^z \binom{z}{k} x^{z-k} y^k$ .

$$\sum_{k=2}^{\varphi} \binom{\varphi}{k} m^{k-1} = \frac{(1 + m)^{\varphi} - 1}{m} - \varphi \quad (3.6)$$

Similarly, for a fixed number of unknowns, the computational complexity of the algorithm is exponential.

## 3.5 Causal Interpretations

Each constraint of a MSO set can be interpreted in a causal way, leading to different causal interpretations. Each interpretation of a constraint is a feasible causal assignment, and is related to the computation of an involved variable, assuming that all the other involved variables are known.

A special notation is used in the incidence matrix to represent the situation when a variable can not be calculated from a constraint. Typically in the literature, an “x” or a “-1” element is used in this situation. When computing a

matching in the incidence matrix, an “x” in the position  $(i, j)$  means that variable  $x_j$  can not be matched with constraint  $c_i$ .

A routine programmed in Matlab<sup>®</sup> finds, if exists, all the different ways in which the set of constraints of each MSO set can be solved. It is based on Algorithms 2 and 4, and finds all the complete matchings for each MSO using the incidence matrix with the calculability information, i.e. information about if a variable can, or can not, be calculated from a constraint. In this way, this routine refines the collection of MSO sets discarding those that do not have a causal interpretation.

The oriented graph for each causal model is obtained by backtracking unknown variables of the unmatched constraint through constraints to which they were matched.

For a MSO set, the different computation sequences leading to residuals are equivalent with respect to the structural detectability/isolability properties (they have the same support). However, they may not be equivalent when considering implementation issues because of, for example, invertibility problems or cycles.

In many cases a constraint can not be solved explicitly for a variable because the inverse of the given function is hard to compute, or even worse, when the constraint is not invertible with respect to a variable, for example, because of a saturation or the use of a lookup table.

Differential constraints can be interpreted either in a derivative way or in an integral way. For the derivative causality, there might be problems, from a numerical point of view, due to the presence of noise when computing the derivative. On the other hand, the initial condition needs to be known to estimate the integral. If there is a preferred causality, this information can be added to the structural model.

Some matchings may lead to cycles or loops in the oriented graph. A cycle is a set of variables mutually dependent, in other words, a set of constraints that have to be solved simultaneously [Düştögör \*et al.\* \(2006\)](#).

There are two kinds of cycles, with different properties: algebraic cycles and differential cycles. An algebraic cycle is made of algebraic constraints only. It is possible to solve the system of equations using algebraic loop solvers that, in general, will attempt to determine the solution iteratively. A differential cycle

represents a system of algebraic and differential equations and the uniqueness of the solution will depend on the context of the problem. As shown in [Blanke \*et al.\* \(2006\)](#), the states of an explicit ordinary differential equation can be solved if the initial conditions of the states are known.

Following a classical graph theory approach, a cycle can be condensed into one single node (which thus represents a subsystem of constraints which need to be solved simultaneously) [Blanke \*et al.\* \(2006\)](#).

A different approach which utilizes equation system solvers, using the characteristics of the algebraic equation solving tools and the differentiating tools, is analyzed in [Svärd & Nyberg \(2008\)](#).

**Example 3.5.1.** *Considering the Example 3.2.1 introduced before, four MSO sets are obtained, and the constraints used in them are shown in Table 3.3.*

Table 3.3: Dependency table for the example.

	$c_1$	$c_2$	$c_3$	$c_4$	$c_5$	$c_6$	$c_7$
$MSO_1$	1	1	1	1	1	1	0
$MSO_2$	1	1	1	1	0	1	1
$MSO_3$	0	0	0	0	1	1	1
$MSO_4$	1	1	1	1	1	0	1

Using the causality information of constraint  $c_4$ , in which only the variable  $x_5$  can be calculated using  $x_1$ ,  $x_2$  and  $x_3$ , two causal models are obtained for  $MSO_1$  and  $MSO_4$ , and three causal models for  $MSO_2$  and  $MSO_3$ . As an example, one oriented graph for each MSO is shown on [Fig. 3.5](#), and the corresponding symbolic expressions of the residual generators are shown in [Eq. 3.7 to 3.10](#).

The symbolic expressions were obtained using a function of the software *SaTool* [Blanke & Lorentzen \(2006\)](#).

$$r_1 \leftarrow c_4(c_1(u_1), c_3(y_1), c_5(y_2), c_2(u_2, c_6(y_3, c_5(y_2)))) \quad (3.7)$$

$$r_2 \leftarrow c_4(c_1(u_1), c_3(y_1), c_2(u_2, c_7(y_4)), c_6(y_3, c_7(y_4))) \quad (3.8)$$

$$r_3 \leftarrow c_6(c_5(y_2), y_3, c_7(y_4)) \quad (3.9)$$

$$r_4 \leftarrow c_5(y_2, c_4(c_1(u_1), c_3(y_1), c_2(u_2, c_7(y_4)))) \quad (3.10)$$

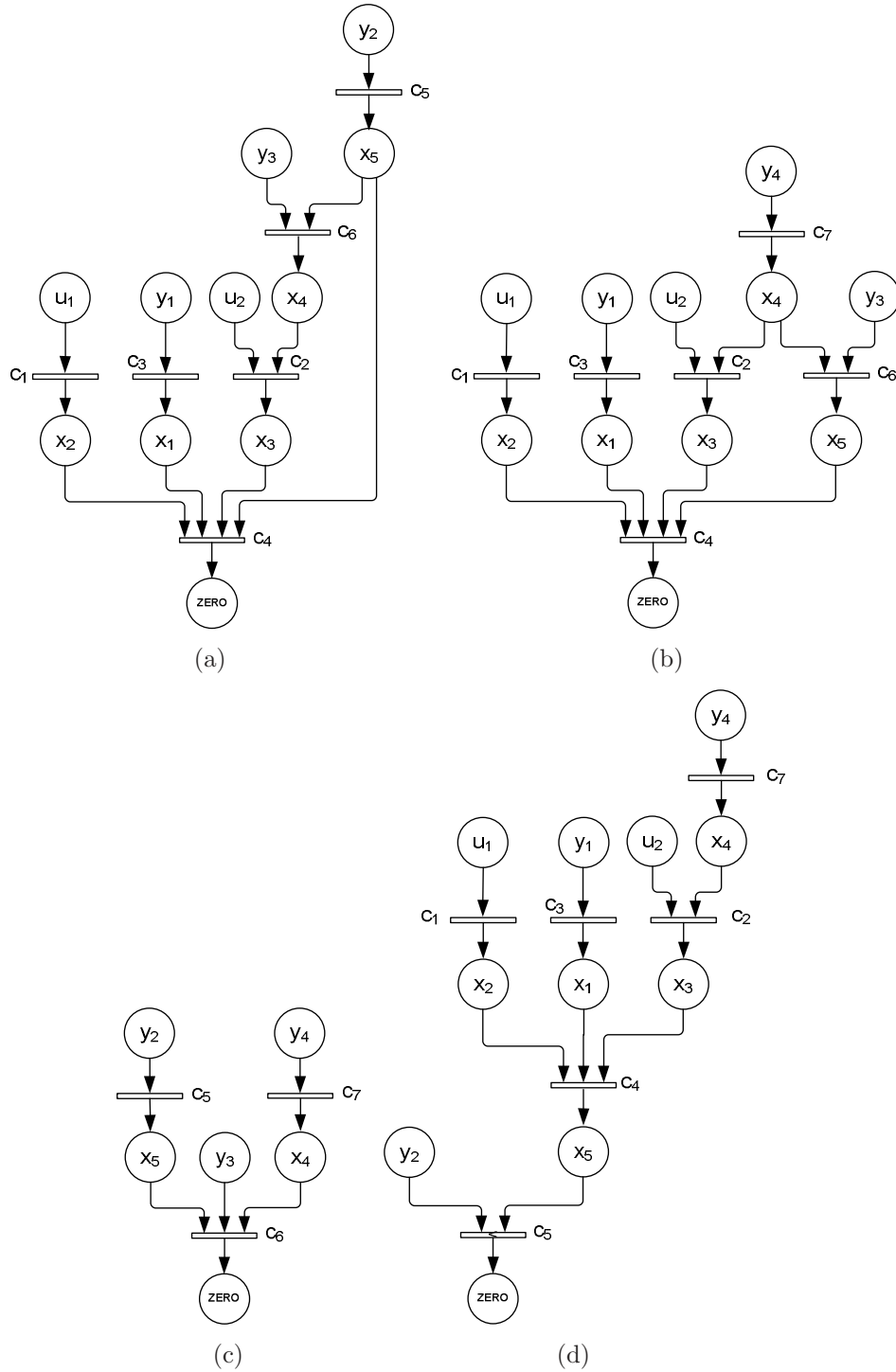


Figure 3.5: Oriented graphs showing the order in which the unknown variables can be determined in (a)  $MSO_1$ , (a)  $MSO_2$ , (c)  $MSO_3$  and (d)  $MSO_4$ .

## 3.6 Structural Detectability and Isolability Analysis

Using the information of the faults which influence each constraint, the (theoretical) Fault Signature Matrix (FSM) can be derived. The fault signature of each possible residual, represents the set of faults which affect at least one of the constraints used to generate this residual. Two other important concepts are defined, such as the support of an ARR, and the scope of a fault.

**Definition 3.6.1.** *The fault signature matrix is a binary table in which each line corresponds to an ARR and each column to a fault. A 0 in the position  $(i, j)$  indicates that the occurrence of the fault  $f_j$  does not affect the ARR  $i$ , and a 1 otherwise [Cordier et al. \(2004\)](#).*

**Definition 3.6.2.** *The support of an ARR,  $ARR_i$ , is the set of faults (columns of the signature matrix which can also correspond to components or constraints) with a nonzero element in the row corresponding to this  $ARR_i$  [Cordier et al. \(2004\)](#).*

**Definition 3.6.3.** *The scope of a fault  $f_j$  is the set of ARRs (rows of the signature matrix) with a nonzero element in the column corresponding to  $f_j$  [Cordier et al. \(2004\)](#).*

The columns of the signature matrix can also be associated with components or constraints, when, for example, a component-centered modeling approach is adopted, and a diagnosis is defined as a set of (faulty) component. Diagnosis abstracted at the component level is typical in approaches coming from the DX community. On the other hand, in the FDI community, columns of the signature matrix are typically associated with variables and parameters of the model which deviate in the presence of a fault [Cordier et al. \(2004\)](#).

The proposed package performs the detectability and isolability analysis of the faults based on the fault signature matrix.

**Definition 3.6.4.** *Structural detectability. A violation of a constraint  $c$  is structurally detectable if and only if it has a nonzero signature in some residual  $r$ , [Blanke et al. \(2006\)](#).*

**Definition 3.6.5.** *Structural isolability. A violation of a constraint  $c_i$  is structurally isolable if and only if it has a unique signature in the residual vector, i.e. respective column  $m_i$  of the FSM is independent of all other columns of the FSM, [Blanke et al. \(2006\)](#).*

### 3.6 Structural Detectability and Isolability Analysis

---

**Example 3.6.1.** *In the Example 3.2.1, all faults are detectable because a violation of any constraint is mapped onto the residuals of the four MSO sets. Considering isolability, as can be deduced from Table 3.3, faults in constraints  $c_5$ ,  $c_6$  and  $c_7$  are structurally isolable. Faults in  $c_1$ ,  $c_2$ ,  $c_3$  and  $c_4$  are group-wise isolable, i.e. within this group individual faults are detectable but not isolable. Table 3.4 shows the structural detectability and isolability of faults in this example.*

Table 3.4: Detectability and isolability in a single fault case.

$c_1$	$c_2$	$c_3$	$c_4$	$c_5$	$c_6$	$c_7$
d	d	d	d	i	i	i

The last definition of structural isolability is based on the ARR-based exoneration assumption Cordier *et al.* (2004), which is common in the FDI approach and follows a column view of the FSM. This assumption establishes that any component in the support of a nonsatisfied ARR is a fault candidate, but also any component in the support of a satisfied ARR is implicitly exonerated (satisfied rows are thus also used in the reasoning). On the other hand, if this exoneration assumption is not taken into account, then the fault candidates can be obtained by the minimal hitting set for the collection of supports of the ARRs which are violated. This approach, typical in the DX approach, follows a row view of the FSM (for further details see Cordier *et al.* (2004)).

**Definition 3.6.6.** *Hitting set. Let  $\mathcal{F} = \{\mathcal{F}_1, \dots, \mathcal{F}_n\}$  be a set of sets, the set  $F \subseteq \bigcup_{\mathcal{F}_i \in \mathcal{F}} \mathcal{F}_i$  is a hitting set of  $\mathcal{F}$  iff*

$$\forall \mathcal{F}_i \in \mathcal{F}, F \cap \mathcal{F}_i \neq \emptyset$$

*A hitting set  $F$  for the set  $\mathcal{F}$  is a minimal hitting set if there is no proper subset  $F' \subset F$ , where  $F'$  is a hitting set for the set  $\mathcal{F}$ .*

In this context, a definition of potential structural isolability is then given as follows.

**Definition 3.6.7.** *Potential structural isolability. A violation of a constraint  $c_i$  may be structurally isolable to a violation of a constraint  $c_j$  if the scope of  $c_i$  is not a subset of the scope of  $c_j$ , i.e. there exists a residual that is sensitive to a violation of  $c_i$  but not  $c_j$ .*

As defined in [Düştögör \*et al.\* \(2006\)](#), in order to easily visualize the isolability property of faults, the isolability matrix is computed.

**Definition 3.6.8.** *The isolability matrix is a square matrix where each row and each column correspond to a fault. A 1 in the position  $(i, j)$  indicates that fault  $i$  is not isolable from fault  $j$ .*

## 3.7 Active Structural Isolation

Besides passive fault diagnosis methods, i.e. the diagnosis is only based on available signals in the system, active fault diagnosis methods can be applied, see e.g. [Campbell \*et al.\* \(2000\)](#); [Niemann \(2006\)](#); [Poulsen & Niemann \(2007\)](#); [Zhang \(1989\)](#). They are based on the inclusion of an auxiliary input signal/vector into the system. It can give a much faster detection compared with a passive fault detection and isolation approach because the auxiliary inputs can be designed to excite specified possible faults with a minimal effect on the complete system [Niemann \(2006\)](#). On the other side, in the passive FDI approach some faults can only be detected when they are excited by disturbance or reference inputs.

In some cases faults are group-wise isolable, i.e within the group individual faults are detectable but not isolable. This does not necessarily imply that isolation can not be achieved in other ways. Indeed, although the same set of residuals will be “fired” when either one or the other of non-structurally isolable constraints is faulty, the time response of the residuals may be different under the different fault cases. Exciting the system with an input signal perturbation may therefore make it possible to discriminate different responses of the same residual set when different constraints within the group are faulty. The active enhancement of isolability was suggested as structural results in [Blanke & Staroswiecki \(2006\)](#), where no algorithms were offered. In this thesis, some algorithms are proposed in the following section to analyze the active structural isolability properties of a system.

The following results originate from [Blanke & Staroswiecki \(2006\)](#):

**Proposition 3.7.1.** *Active input to residual isolation is possible if and only if both a structural condition and a quantitative condition are true.*  
*Structural condition: the known variables in the set of residuals associated with*

a group of nonstructurally isolable constraints include at least one control input. Quantitative condition: the transfer from control inputs to residuals is affected differently by faults on different constraints.

Observing that input to output structural isolability and input to residual structural isolability are similar with respect to graph properties, we have

**Proposition 3.7.2.** *Active input to output structural isolation is possible if and only if both a structural condition and a quantitative condition are true.*

*Structural condition: the known variables in the set of residuals associated with a group of nonstructurally isolable constraints include at least one control input. Quantitative condition: the transfer from control inputs to outputs is affected differently by faults on different constraints.*

Two lemmas for structural reachability and active structural isolability from [Blanke & Staroswiecki \(2006\)](#) are employed to arrive at constructive algorithms for input to output structural isolability.

**Lemma 3.7.1.** *Input to output structural reachability. Let  $p^{(i,j)} = \{c_f, c_g, \dots, c_h\}$  be a path through the structure graph from input  $u_i$  to output  $y_j$ , and  $\Pi^{(i,j)}$  the union of valid paths from  $u_i$  to output  $y_j$ . Let  $C_{reach}^{(i,j)} = \{c | c \in \Pi^{(i,j)}\}$ . A constraint  $c_h$  is input reachable from input  $u_i$  if a path exists from input  $u_i$  to output  $y_k$  and the path includes the constraint,  $c_h \in C_{reach}^{(i,k)}$ .*

**Lemma 3.7.2.** *Active structural isolability. Two constraints  $c_g$  and  $c_h$  are active structurally isolable if*

$$\exists i, j, k, l : c_g \in C_{reach}^{(i,j)}, c_h \in C_{reach}^{(k,l)}, \text{ and } \{c_g, c_h\} \notin C_{reach}^{(i,j)} \cap C_{reach}^{(k,l)} \quad (3.11)$$

### 3.7.1 Algorithms

The following algorithms consider input to output structural isolability. The first algorithm determines all paths through the structure graph from each input to each output, considering also loops in the system. The input of Algorithm 7, which searches recursively each path by using Algorithm 8, is the adjacency matrix of the directed graph using the signal flow information. Algorithm 8 follows a depth-first search strategy ([Cormen et al. \(2003\)](#)) to traverse the graph until a goal node is found. The search remembers previously-visited nodes and will not repeat them to find forward or simple paths. This algorithm is also used to find a loop, i.e. a path which ends at the node where it began.



---

**Algorithm 7** Algorithm to find the reachability matrix

---

**Input:** adjacency matrix  $A$ , source vertexes  $V_s$ , destination vertexes  $V_d$

- 1: **for all** vertex  $v$  in  $A$  **do**
- 2: find loops using `find_all_paths( $v, v$ )`;
- 3: **end for**
- 4: **for all** vertex  $v_s$  in  $V_s$  and vertex  $v_d$  in  $V_d$  **do**
- 5:  $all\_paths := \text{find\_all\_paths}(v_s, v_d)$ ;
- 6:  $\Pi^{(v_s, v_d)}$ , the reachability from  $v_s$  to  $v_d$ , is equal to constraints in ( $all\_paths$  + loops in  $all\_paths$ );
- 7: **end for**
- 8: **return** reachability matrices

---



---

**Algorithm 8** `find_all_paths` function

---

**Input:** current vertex  $v$ , destination vertex  $v_d$

- 1: mark  $v$  as visited;
- 2: **for all** vertex  $i$  adjacent to  $v$  such that  $i$  not visited **do**
- 3: **if**  $i$  is equal to  $v_d$  **then**
- 4: add to the set of paths;
- 5: **else**
- 6: `find_all_paths( $i, v_d$ )`;
- 7: **end if**
- 8: **end for**
- 9: **return** set of paths

---

**Example 3.7.1.** Consider Example 3.2.1 introduced before, but now with the signal flow causality shown in Fig. 3.6.

The set of paths through constraints from  $u_1$  to the outputs can be found by applying Algorithm 7, and this information is summarized in Table 3.5 of reachability. For example, the following path between  $u_1$  and  $y_3$  is found by using Algorithm 8:

$$u_1 - c_1 - x_2 - c_4 - x_5 - c_6 - y_3$$

The reachability from  $u_2$  is shown in Table 3.6.

Following Lemma 3.7.2, it can be deduced that  $\{c_1, c_2, c_4, c_5, c_6, c_7\}$  are structurally isolable when active isolation is employed, while  $c_3$  is not detectable, as it is shown in Table 3.7.

### 3.7 Active Structural Isolation

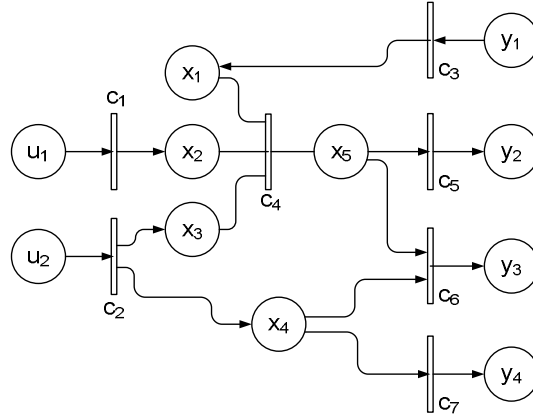


Figure 3.6: Structure graph of Example 3.7.1.

Table 3.5: Output reachability from  $u_1$ .

$u_1 \downarrow$	$c_1$	$c_2$	$c_3$	$c_4$	$c_5$	$c_6$	$c_7$
$y_1$	0	0	0	0	0	0	0
$y_2$	1	0	0	1	1	0	0
$y_3$	1	0	0	1	0	1	0
$y_4$	0	0	0	0	0	0	0

Table 3.6: Output reachability from  $u_2$ .

$u_2 \downarrow$	$c_1$	$c_2$	$c_3$	$c_4$	$c_5$	$c_6$	$c_7$
$y_1$	0	0	0	0	0	0	0
$y_2$	0	1	0	1	1	0	0
$y_3$	0	1	0	1	0	1	0
$y_4$	0	1	0	0	0	0	1

Table 3.7: Active structural detectability and isolability of some single faults.

$c_1$	$c_2$	$c_3$	$c_4$	$c_5$	$c_6$	$c_7$
i	i	-	i	i	i	i

### 3.8 A Case of Multiple Modes of Operation

Talking about the operation of a plant, two concepts could be distinguished, the state, and the operation mode or protocol.

The state of a plant is defined as a set of operation parameters that determine the factory operation, together with a set of medium- to long-term restrictions on these parameters, on some plant variables and even on plant configuration [Acosta \*et al.\* \(2002\)](#). Each state of the plant could be associated with one or more operation modes or protocols.

The operation mode includes, for example, automatic switches and/or processes that have a different behavior depending on the operating range. The equational model of such systems is formed of a set of equations among which some have associated conditions defining their operation range [Travé-Massuyès & Pons \(1997\)](#). These hybrid systems exhibit both continuous and discrete dynamic behavior.

The equations of the model may have an associated condition,  $C$ , which take truth values according to the satisfaction of the condition. Hence, an equation can have no condition or can be submitted to a condition  $C$  or to  $\neg C$ .

The conditions defining the different modes of some physical component (represented by one or several equations) must define a partition of the parameter subspace spanned by the parameters appearing in these conditions. i.e. they must be mutually exclusive and cover the whole subspace [Travé-Massuyès & Pons \(1997\)](#).

The number of different operation modes for the system can be obtained by multiplying the number of different modes for every physical component [Travé-Massuyès & Pons \(1997\)](#). However, some of the combinations of operation modes of the components may be not valid because they may not be physically possible or they may not be considered in the normal operation of the system. It is important to take into account only the valid ones when computing the MSO sets to reduce the computational load.

In the first stage of this thesis work and under the supervision of Prof. Belarmino Pulido, the MSO sets and their causal models are obtained in an incremental way when conditions are considered in the constraints. A modification

### 3.8 A Case of Multiple Modes of Operation

of the Possible Conflict Computation (PCC) algorithm from Pulido & Alonso-González (2004) is used. The PCC algorithm uses a Depth First Search and, because of this, the conditions of the equations are analyzed while a MSO set or a causal model is built. A description of this work and the pseudocode of the modified algorithm can be found in Pulido & Gelso (2005).

The structural isolability of a fault can be improved using information from different operation modes. Different residual generators may be obtained in different operation modes, and hence, different detectability and isolability properties of the faults. Combining this information and under the ARR-based exoneration assumption (see Section 3.6), a fault which is only detectable in any operation mode may have now a unique signature, and in this manner the fault can be isolable. In Laursen *et al.* (2008), the improvement of the structural isolability is demonstrated on a water for injection distribution process working in different use-modes.

**Example 3.8.1.** *The illustrative system of Fig. 3.7 includes a voltage source  $V_s$ , a resistor  $R1$ , two lights  $Light1$  and  $Light2$ , two switches  $S1$  and  $S2$ , and three meters, voltmeters  $V1$  and  $V2$ , and ammeter  $A$ . The switches and the meters are considered as ideal ones, i.e. a switch has no voltage drop across it during its ON state and it has infinity resistance during its OFF state, and the measuring instruments do not disturb the circuit.*

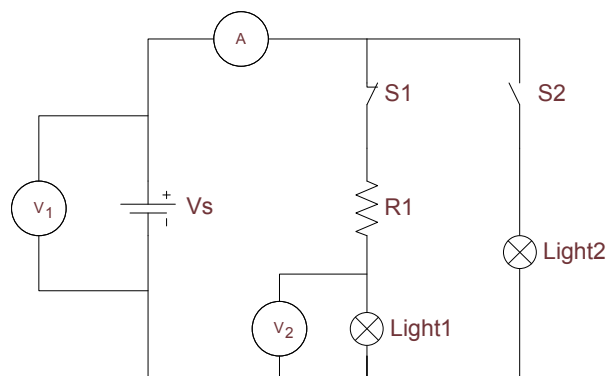


Figure 3.7: An electric circuit with multiple operation modes.

$C_1$  is the condition that the switch  $S1$  is “closed”, and  $C_2$  is the condition that the switch  $S2$  is “closed”. Assume that two operation modes are considered in the normal operation of this example. In the first, the switch  $S1$  is closed and the

### 3.8 A Case of Multiple Modes of Operation

switch  $S2$  is open. In the second, both switches are closed. The relations between the conditions and the operation modes are represented in Table 3.8.

Table 3.8: Relations between the conditions and the operation modes.

$\neg C_1$	$\neg C_2$	Not valid	$S1$ : open	$S2$ : open
$\neg C_1$	$C_2$	Not valid	$S1$ : open	$S2$ : closed
$C_1$	$\neg C_2$	Operation mode 1	$S1$ : closed	$S2$ : open
$C_1$	$C_2$	Operation mode 2	$S1$ : closed	$S2$ : closed

The equations are the following:

$$\begin{array}{ll}
 e_1 & \tilde{V}_1 = V_1 \\
 e_2 & \tilde{V}_2 = V_2 \\
 e_3 & \tilde{I} = I \\
 e_4 & I = I_1 + I_2 \\
 e_5 & I_1 = 0 \qquad \qquad \qquad \neg C_1 \\
 e_6 & I_2 = 0 \qquad \qquad \qquad \neg C_2 \\
 e_7 & V_1 = V_{R_1} + V_2 \qquad \qquad C_1 \\
 e_8 & V_{R_1} = I_1 R_1 \qquad \qquad C_1 \\
 e_9 & V_2 = I_1 R_{Light1} \qquad \qquad C_1 \\
 e_{10} & V_1 = I_2 R_{Light2} \qquad \qquad C_2
 \end{array}$$

Four and five MSO sets are obtained for the operation modes 1 and 2, respectively (see Table 3.9).

Table 3.9: MSO sets of operation modes 1 and 2.

Operation mode 1		Operation mode 2	
$n^o$	MSO sets	$n^o$	MSO sets
1	$\{e_2, e_3, e_4, e_6, e_9\}$	1	$\{e_1, e_2, e_3, e_4, e_9, e_{10}\}$
2	$\{e_1, e_2, e_7, e_8, e_9\}$	2	$\{e_1, e_2, e_7, e_8, e_9\}$
3	$\{e_1, e_2, e_3, e_4, e_6, e_7, e_8\}$	3	$\{e_1, e_2, e_3, e_4, e_7, e_8, e_{10}\}$
4	$\{e_1, e_3, e_4, e_6, e_7, e_8, e_9\}$	4	$\{e_2, e_3, e_4, e_7, e_8, e_9, e_{10}\}$
		5	$\{e_1, e_3, e_4, e_7, e_8, e_9, e_{10}\}$

Table 3.10 shows the structural detectability and isolability of faults in the sensors, lights and the resistance. As can be seen in the table, the structural

*isolability is improved using both operation modes. In each operation mode some faults are only detectable, whereas all faults are isolable when the information from both operation modes are combined. For example, a fault in Light2 is only detectable in the second operation mode, but combining the information of both modes it has a unique signature, and hence, is isolable.*

Table 3.10: Detectability and isolability using both operation modes.

	$V_1$	$V_2$	$A$	<i>Light1</i>	<i>Light2</i>	$R_1$
Mode 1	d	i	i	i	-	d
Mode 2	i	i	d	i	d	i
Both	i	i	i	i	i	i

## 3.9 Conclusions and Contributions

One of the main contributions of this chapter is the proposed algorithm to obtain the whole set of MSO sets of the system. When talking about over constrained systems, the task of obtaining the complete set of minimal mathematical relations (to retrieve an optimal diagnostic) becomes a complex task.

As it is shown in this chapter, the proposed algorithm become one of the alternatives to perform the task of designing fault detection and isolation tests, by finding all the subsystems that could be diagnosed in the system. This algorithm completes previous approaches, as e.g. explained in [Blanke & Lorentzen \(2006\)](#), which do not deliver all the MSO sets, and finds them in an efficient way without investigating all different matchings, as e.g. in [Laursen et al. \(2008\)](#). Moreover, calculability in the constraints can be taken into account to remove MSO sets that can lead to unachievable detection tests.

Features of this algorithm were presented and its efficiency is demonstrated by several application examples of Chapter 7. The computational complexity of the proposed algorithm is polynomial in the number of unknowns making this condition suitable for real-world systems with a large number of unknown variables and constraints.

Structural detectability and isolability of faults could be taken into account from (i) a passive fault diagnosis approach, by using the set of all feasible MSO

### 3.9 Conclusions and Contributions

---

sets and the corresponding fault signature matrix, and (ii) an active fault diagnosis approach, by using an auxiliary input in the system and the transfer function from the auxiliary input to the residual output. Both approaches were addressed from a structural point of view. After applying them to examples, different structural detectability and isolability properties were obtained, making possible to use the passive and the active techniques to enhance fault isolation.

Finally, the algorithm proposed in this thesis to find all the MSO sets is compared with other three algorithms which are representative for the domain. This comparison included the computational complexity, and the correctness of the solution provided when applying to several case studies. *Armengol et al. (2009a)* summarizes part of the results.

# Chapter 4

## Robust Fault Detection based on Interval Models

### Summary

In the previous chapter, the way in which the structural analysis can be used to deduce the analytical redundancy relations was presented. An important difficulty in applying these fault tests to real systems is dealing with the uncertainties associated with the system itself and with the measurements.

In order to deal with these uncertainties, during the development of this thesis, mainly two interval approaches were used. The former is the SQualTrack, which is a software package to detect faults based on Modal Interval Analysis. The latter, Interval-based Consistency Techniques - ICTs, is based on “classic” interval analysis and constraint propagation techniques. Related to the SQualTrack, two improvements are proposed to increase the fault detection performance. Concerning the ICTs, the work developed during this thesis includes: (i) state the fault detection problem as a constraint satisfaction problem, and (ii) design and develop a fault detection system using the solver called RealPaver.

Finally, the ICTs fault detection performance is empirically compared with the performance of SQualTrack and a statistical-decision technique: Extended Kalman filter and Z-test. This empirical comparison was developed during a research stay at Vanderbilt University (Nashville-USA) under the supervision of Prof. Gautam Biswas.



## 4.1 Related Work

Over the last years, many works using interval techniques for fault detection have appeared. One of the main drawbacks is the high computational load that they require at each sampling time, which in general is incompatible for working with on-line data. Some approaches overcame this problem, as for example in

- Fagarasan *et al.* (2004); Ploix & Gentil (2000), working with simple multiple-input single-output models to generate the exact estimate of the output using interval calculation laws.
- Rinner & Weiss (2004), where the bounds of the estimates are computed using traditional numerical integration techniques from the uncertain parameter interval vertices, assuming that the monotonicity property (of the state values with regard to the parameters) holds.

The methods that use interval arithmetic Moore (1966), in the general case for example using the natural extension, do not provide the exact result (complete and sound). This comes essentially from the multi-incidences and the wrapping effect Armengol *et al.* (2001a). Interval arithmetic provides overbounded results for the range of functions when there are multi-incident variables or there are dependencies or relations between variables. On the other hand, the wrapping problem appears when interval arithmetic is used for simulation in the state space. The state, at any time point, of an interval model is enclosed by a  $n$ -dimensional hypercube, hence usually includes spurious states.

An alternative to classical Interval Analysis (IA) is the Modal Interval Analysis (MIA) Gardenyes *et al.* (2001); SIGLA/X group (1999). It is a completion of the Interval Analysis, not only in a lattice and arithmetic sense, as the Extended Intervals of Kaucher Kaucher (1979), but a logical completion as well. A modal interval  $X$  is defined as a couple  $X = (X', \forall)$  or  $X = (X', \exists)$  where  $X'$  is its classic interval *domain*,  $X' \in I(\mathbb{R})$ , and the quantifiers  $\forall$  and  $\exists$  are a selection *modality*. MIA allows in some cases to reduce the overestimation that classical IA produces when evaluating an interval function because of the problem of multi-incident variables. The software called SQualTrack Armengol *et al.* (2003, 2009b) combines MIA and branch-and-bound techniques to compute simultaneously an

underbounded estimate and an overbounded one. This software is described in this chapter.

Most of the works that use an interval model of a dynamic system are in discrete-time. However, [Lin & Stadtherr \(2007, 2008\)](#) proposed a method for fault detection using continuous-time models described by ordinary differential equations (ODE) with interval-valued parameters and/or initial states. The implementation, based on a simulation of Taylor models, produces a complete but not sound estimate. A constraint propagation procedure and the output measurements are used to accelerate the fault detection process by reducing, at each time step, the uncertainties in the model parameters and initial states.

Set-membership identification methods have also been used for robust fault detection. They explicitly calculate outer bounds of the set of parameters that are consistent with the measurements. They can be classified according to how the approximation of the feasible set of parameters is represented or parameterized: by ellipsoids in [Watkins & Yurkovich \(1996\)](#), by parallelotopes in [Ingimundarson \*et al.\* \(2005\)](#), by polytopes in [Janati-Idrissi \*et al.\* \(2002\)](#), and by zonotopes in [Combastel \*et al.\* \(2008\)](#); [Guerra \*et al.\* \(2006, 2007\)](#); [Ingimundarson \*et al.\* \(2009\)](#); [Manders \(2008\)](#). In general, the methods deal with models that are linear in the parameters. The zonotope-based fault detection algorithm presented in [Ingimundarson \*et al.\* \(2009\)](#) is able to handle invariant parameters, parameter variation bounded between samples, and unbounded parameter variation.

Constraint propagation techniques tighten the domains of variables involved in constraint systems. In order to separate the solutions, they are generally embedded in a bisection algorithm that iteratively prunes by propagation the domains split into sub-domains if the required precision is not yet reached [Granvilliers \*et al.\* \(1999\)](#). [Gelso \*et al.\* \(2007c\)](#); [Stancu \*et al.\* \(2003\)](#) have applied these techniques using bisection and the solvers RealPaver and Project2D, respectively, and [Ocampo-Martínez \*et al.\* \(2006\)](#) have used constraint propagation techniques without bisections by using the IntervalPeeler Solver. These techniques are presented in detail in this chapter.

In [Puig \*et al.\* \(2008\)](#), three schemes (simulation, prediction, and observation) are compared depending on the value of the gain of an interval observer of a linear dynamic system in discrete-time. As a summary of main differences, it can be said

that prediction and simulation approaches have antagonist properties: prediction does not suffer from the wrapping effect, has low computational complexity, has low sensitivity to unmodeled dynamics but suffers the “fault following effect” (the predicted output tends to follow the faulty system output), and has high sensitivity to sensor noise. On the other side, the simulation approach has the opposite properties. Observer approach is in the middle, with the advantage that since it has one more degree of freedom (the observer gain), it can be designed trying to minimize the bad effects and maximize the good effects. An interval observer is presented in Section 4.3.1.1 and applied to one application example in Chapter 7.

## 4.2 A Tool based on Modal Intervals: SQualTrack

The model based fault detection using modal intervals, is based on the calculation of effective thresholds (*adaptive thresholds* or *envelopes*) to bound the uncertainty of parameters and measurements. To compute the envelope limits, it is necessary to compute the range of a function in a given parameter space at each prediction step, which is very costly. SQualTrack [Armengol \*et al.\* \(2003, 2009b\)](#) calculates iteratively external estimates

$$\hat{Y}_{ex}(k) \supseteq \hat{Y}(k), \quad (4.1)$$

of the predicted behavior from the model  $\hat{Y}(k)$ , so the cost is heavily reduced. These estimates are closer at each iteration and, after an infinite number of iterations, the exact range  $\hat{Y}(k)$  would be calculated. But the algorithm stops when the external estimate is close enough to detect the fault, thus saving much computational effort for the detection of faults. However, if no fault is detected, the algorithm will never stop. This drawback can be overcome by using an internal estimate,

$$\hat{Y}_{in}(k) \subseteq \hat{Y}(k), \quad (4.2)$$

which is included in the exact envelope. If the measurement is within this envelope, then the fault, if it exists, will not be detected, and so the algorithm will

stop iterating. The internal and external estimates of the exact envelope, which are depicted in Fig. 4.1, define three zones.

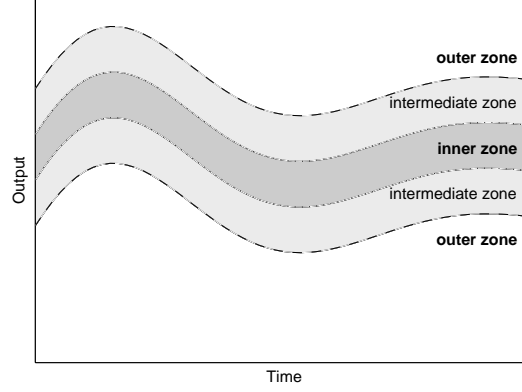


Figure 4.1: The three zones defined by the internal and external estimates of the exact envelope.

Therefore a fault is detected when the measured value is either larger or smaller than the predicted value by the external estimate or, in other words, when the output of the model is not consistent with the measured output. This assertion is expressed through the logical statement,

$$\begin{aligned} & \neg((\exists \tilde{\mathbf{y}}(k) \in \tilde{\mathbf{Y}}(k)) (\exists \hat{\mathbf{y}}(k) \in \hat{\mathbf{Y}}(k)) \tilde{\mathbf{y}}(k) - \hat{\mathbf{y}}(k) = \mathbf{0}) \Leftrightarrow \\ & (\forall \tilde{\mathbf{y}}(k) \in \tilde{\mathbf{Y}}(k)) (\forall \hat{\mathbf{y}}(k) \in \hat{\mathbf{Y}}(k)) \mathbf{r}(k) \neq \mathbf{0}. \end{aligned} \quad (4.3)$$

where  $\tilde{\mathbf{y}}(k)$  is the measured output of the system at instant  $k$ ,  $\hat{\mathbf{y}}(k)$  is the predicted output of the model at instant  $k$ , and  $\mathbf{r}(k)$  is the so-called residual of the analytical redundancy relation (ARR) of the system.

Actually, the consistency between the interval model and the real process is performed using interval measurements, which are obtained from the measurements taking into account the uncertainties (noise, bias...) of the sensors.

The use of the Modal Interval Analysis (MIA) [SIGLA/X group \(1999\)](#) in the fault detection system guarantees that a fault exists when the measurement is out of the external envelope (in the outer zone), so this method does not generate false alarms. If there are false alarms, they indicate either that the interval model does not represent the system adequately, or that the interval measurements do

not represent the true values of the variables adequately. On the other side, if the measurement is in the intermediate zone or in the inner zone there can be missed alarms.

The  $f^*$  algorithm is used to compute the internal and external approximations in an iterative way [Herrero \(2006\)](#). It combines Modal Interval Analysis and branch-and-bound techniques. The basic concept consists of bisecting in a intelligent manner the variable space by studying the monotonic nature of the function with respect to each of its variables. Then, by applying some theorems from Modal Interval Analysis, which are also based on the monotony study, it is possible to obtain better results in the evaluation.

Any measurement belonging to a past time point can be used as initial state to compute the envelopes at the current time point. The time interval from this initial time point to the current one is called *time window*. If the window used at each prediction step has always the same length, then a *sliding time window* is being considered [Armengol et al. \(2001b\)](#).

Then, for a window of length  $w$ , in accordance with Equation 4.3, a system is faulty if

$$(\forall \tilde{\mathbf{y}}(k) \in \tilde{\mathbf{Y}}(k))(\forall \hat{\mathbf{y}}(k) \in \hat{\mathbf{Y}}(k|k-w)) \tilde{\mathbf{y}}(k) - \hat{\mathbf{y}}(k|k-w) \neq 0, \quad (4.4)$$

where, for example, for a discrete first-order model

$$\hat{\mathbf{y}}(k|k-w) = \mathbf{f}(\tilde{\mathbf{y}}(k-w), \tilde{\mathbf{u}}(k-1), \dots, \tilde{\mathbf{u}}(k-w), \mathbf{p}) \quad (4.5)$$

$\mathbf{p}$  is the vector of uncertain model parameters that belongs to the vector of intervals  $\mathbf{P}$ . Two situations must be distinguished in this case. A first case is when the system is considered to be time variant, in which the values of  $\mathbf{p}$  at different time steps can be considered different. A second case is when the system is considered time invariant, i.e., there is uncertainty in the values of  $\mathbf{p}$  but it is known that these values are the same at different time steps. This second case, time invariant systems, is more difficult to handle than the first one because of the multi-incidences of  $\mathbf{p}$  in the expression of  $\hat{\mathbf{y}}(k|k-w)$ . The time invariant case is generally considered when the SQualTrack is used.

The number of missed alarms is in general reduced by using several window lengths simultaneously, as a fault is detected when there is an inconsistency in

a time window. Moreover, the best window length not only depends on the dynamics of the system but also on the type of fault to be detected.

**Example 4.2.1.** *Given the ARR corresponding to a discrete first-order model [Herrero \(2006\)](#),*

$$\hat{y}(k) = a\tilde{y}(k-1) + bu(k-1), \quad (4.6)$$

where  $a$  and  $b$  are model parameters, in this case, time invariant within the time window. The corresponding ARRs for the set of window lengths  $\mathbf{w} = \{1, 5\}$  are,

$$\text{Length } w = 1 : \quad \hat{y}(k) = a\tilde{y}(k-1) + bu(k-1), \quad (4.7)$$

$$\begin{aligned} \text{Length } w = 5 : \quad \hat{y}(k) = & a(a(a(a(a\tilde{y}(k-5) + bu(k-5)) + \\ & bu(k-4)) + bu(k-3)) + bu(k-2)) + bu(k-1). \end{aligned} \quad (4.8)$$

Therefore, proving that either [Equations 4.9](#) or [4.10](#) are true is equivalent to say that a fault is detected.

$$\begin{aligned} \text{Length } w = 1 : \quad & (\forall \tilde{y}(k) \in \tilde{Y}(k))(\forall \tilde{y}(k-1) \in \tilde{Y}(k-1)) & (4.9) \\ & (\forall u(k-1) \in U(k-1))(\forall a \in A)(\forall b \in B) \\ & \tilde{y}(k) - (a\tilde{y}(k-1) + bu(k-1)) \neq 0 \end{aligned}$$

∨

$$\begin{aligned} \text{Length } w = 5 : \quad & (\forall \tilde{y}(k) \in \tilde{Y}(k))(\forall \tilde{y}(k-1) \in \tilde{Y}(k-1)) & (4.10) \\ & (\forall u(k-1) \in U(k-1))(\forall u(k-2) \in U(k-2)) \\ & (\forall u(k-3) \in U(k-3))(\forall u(k-4) \in U(k-4)) \\ & (\forall u(k-5) \in U(k-5))(\forall a \in A)(\forall b \in B) \\ & \tilde{y}(k) - (a(a(a(a(a\tilde{y}(k-5) + bu(k-5)) + bu(k-4)) + \\ & bu(k-3)) + bu(k-2)) + bu(k-1)) \neq 0. \end{aligned}$$

In the following, two improvements of SQualTrack proposed in this thesis are presented. The former refers to the dynamic refinement of the parameters space, and the latter, to the pruning of the measurements space. The impact that these improvements might take in the SQualTrack performance is illustrated by a chemical plant. Although both improvements can be applied in a simultaneous way, for a sake of clarity, they are applied separately to the same example.

### 4.2.1 Dynamic Refinement of the Parameters Space

The dynamic refinement of the parameters space is based on the partitioning of the space into several subspace models. At each time step, the consistent subspaces are partitioned into smaller subspaces and the inconsistent ones are refuted and excluded. In this way, a fault is detected when all subspaces are refuted. In this analysis it is assumed that the parameters do not vary over time.

A partition is defined as

$$Q = [(\underline{q}_1, \overline{q}_1), (\underline{q}_2, \overline{q}_2), \dots, (\underline{q}_n, \overline{q}_n)] \quad (4.11)$$

with  $Q \subseteq P$ , being  $P$  the interval vector of parameters to be divided. Thus, a partition divides the uncertainty space into smaller regions. The union of all subspace models covers the complete (initial) uncertainty space of the imprecise model [Rinner & Weiss \(2004\)](#):  $\bigcup_m Q^{(m)} = P$  where  $m = 1, \dots, M$ .

A methodology for parameter estimation explained in [Castillo \(2007\)](#); [Castillo et al. \(2007\)](#) has been used as starting point for the implementation of this improvement. This technique divides consecutively the interval for each parameter, and checks the satisfiability of the constraints of each resulting subspace using the Quantified Real Constraint Satisfaction (QRCS) solver [Herrero \(2006\)](#). Fig. 4.2 shows the procedure used to partition the interval of one parameter, and thus, to find the new bounds of it. In the same figure,  $T$  (true) and  $F$  (false) indicates the result of applying the QRCS solver to the constraints of a particular subspace. There is a trade-off between the number of partitions (and hence, callings to the QRCS solver) and the computational load.

In [Rinner & Weiss \(2004\)](#), a system for dynamically refining imprecise models is presented. It is also based in the dynamic partitioning of the parameters space, but instead of using interval arithmetic, numerical integration techniques are used to calculate the solution of the differential equations of the model. A monotonicity condition is needed because, if it is not given, underbounded envelopes are computed. In that case, the parameter refinement is not performed. The monotonicity assumption is not required in the method proposed in this thesis.

## 4.2 A Tool based on Modal Intervals: SQualTrack

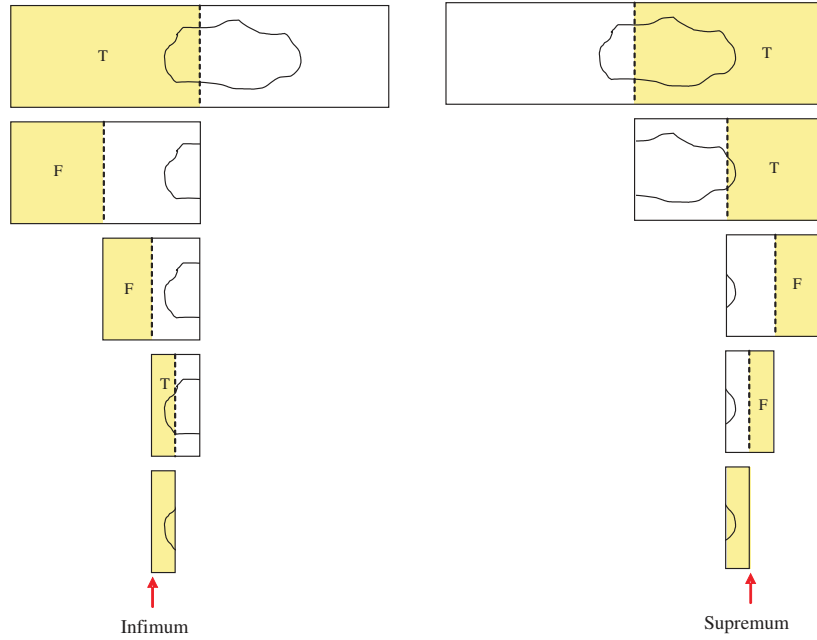


Figure 4.2: Procedure used to find the new bounds of one parameter.

**Example 4.2.2.** *As a simple illustration of the method, consider a chemical tank equipped with a heat exchanger. Figure 4.3 shows the schematic of this system: a tank, a heat exchanger, and two pumps.*

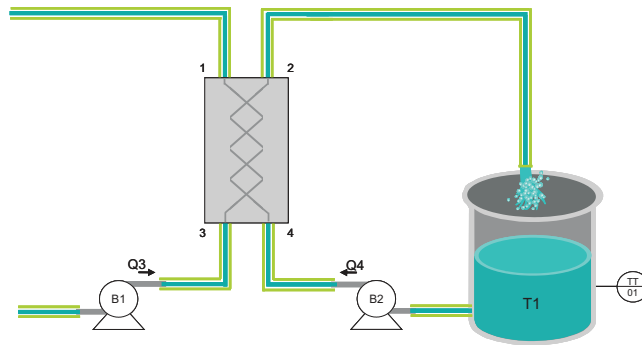


Figure 4.3: Scheme of the chemical tank equipped with a heat exchanger

*The analytical model of the above system can be obtained using basic rules in thermodynamics. Employing those rules for both tank and heat exchanger and after some manipulation on equations, a discrete analytical model of the system can be written like this*



## 4.2 A Tool based on Modal Intervals: SQualTrack

---

$$T_4[k + 1] = T_4[k] \left[ 1 - \frac{T_s}{V} (A + B) \right] + \frac{T_s}{V} (A * T_3[k] + B * T_a) \quad (4.12)$$

where  $T_3$  and  $T_4$  are temperatures in points 3, and 4 respectively;  $V$  is the volume of water in tank;  $T_a$  is the environmental temperature considered unknown but bounded in simulations; and  $T_s$  is the sample time.  $A$  and  $B$  parameters are denoting for:

$$A = \frac{1}{\frac{1}{Q_3} + \frac{1}{Q_4}} \quad (4.13)$$

$$B = \frac{K}{Ro * C} \quad (4.14)$$

where  $Ro$ ,  $C$  and  $K$  are density, specific heat capacity, and heat transfer rate of the exchanger respectively. The considered intervals for parameters in the analytical model and their dimensions have been listed in Table 4.1. In simulations, sample time is chosen 5 s without any concern about losing any part of output profile containing some meaningful deviation. Initial temperature of water in tank is 60 °C.

Table 4.1: Parameter intervals of the chemical plant example.

$V = [1, 1.1] \text{ m}^3$	$T_a = [19.8, 20.2] \text{ }^\circ\text{C}$
$A = [0.001, 0.011] \frac{\text{m}^3}{\text{s}}$	$B = [0.001, 0.011] \frac{\text{m}^3}{\text{s}}$

The fault scenario that is considered is a fault in pump B1, a decrement of the pump performance, which occurs at sample 130.

Figure 4.4 displays the results of fault detection using SQualTrack, without the refinement of the parameters space, and a time window of length 8. The upper graph shows the approximations (inner and outer) for the output variable together with the measured output, which is plotted with a black solid line. The lower graph shows a bar when a fault is detected, i.e., when the intersection between the interval measurement and the external approximation is void. As it can be seen in the figure, the fault is not detected.

Figure 4.5 displays the results for the same scenario, but now including the refinement of the parameters space. In this case, the fault is detected from sample 299.

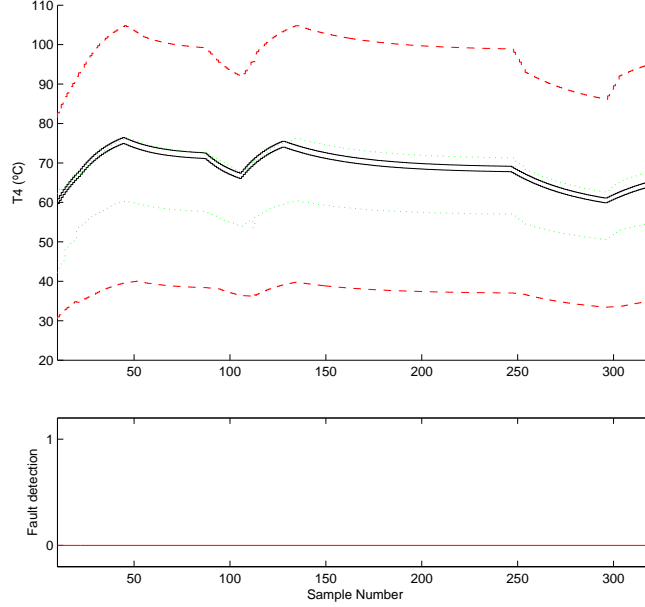


Figure 4.4: A fault in pump *B1* using SQualTrack, without the refinement of the parameters space.

### 4.2.2 Dynamically Refining the Measurements Space

One way to prevent the propagation of uncertainty is to prune the measurements space. This pruning corresponds to an update of the measurement of the output of the system since only it is stored the measurement space consistent with the output estimated by the interval model.

The updated measurement  $\tilde{Y}^*(k)$  is calculated by means of Equation 4.15.

$$\tilde{Y}^*(k) = \tilde{Y}(k) \cap \hat{Y}(k/k - w_i) \quad (4.15)$$

Where  $w_i$  represents the length of the sliding time window  $i$ . In this way, the propagation of the uncertainty not consistent with the model of the system may decrease when  $\tilde{Y}^*(k)$  is used as the initial value of a sliding time window.

## 4.2 A Tool based on Modal Intervals: SQualTrack

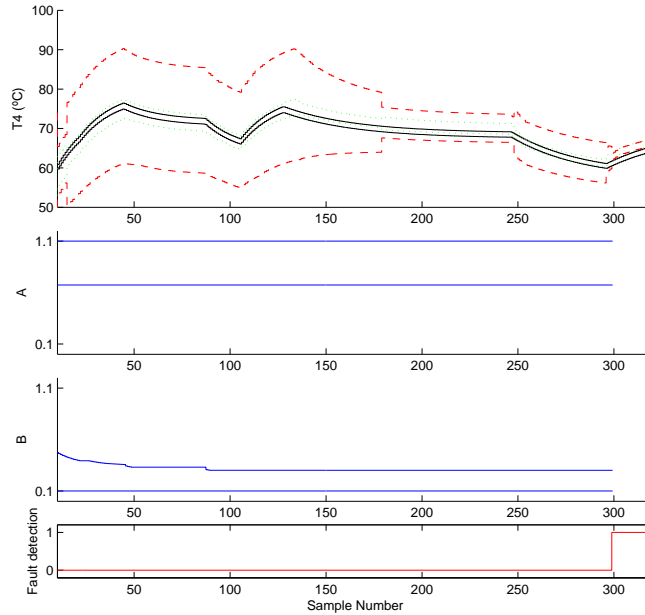


Figure 4.5: A fault in pump  $B1$  using SQualTrack, with the refinement of the parameters space.

**Example 4.2.3.** *As an example of the improvement of the fault detection performance, the chemical tank and the heat exchanger introduced before are considered. The intervals for parameters  $A$  and  $B$  in the analytical model are as the following,  $A = [0.0095, 0.015]$  and  $B = [0.0015, 0.0018]$ .*

*The faulty scenario consists of a fault in pump  $B1$ , a decrement of the pump performance, which begins at sample 130.*

*Figure 4.6 displays the results of fault detection using SQualTrack, without the refinement of the measurements space. As it can be seen in the figure, the fault is not detected.*

*The results for this scenario, under the same conditions but pruning the measurement space, are shown in Figure 4.7. In this case the fault is detected from sample 205.*

### 4.3 Fault Detection as a Constraint Satisfaction Problem

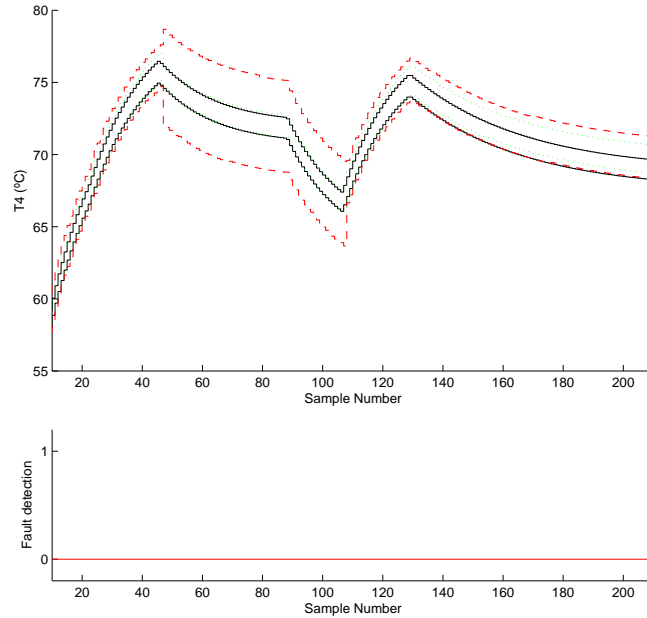


Figure 4.6: A fault in pump  $B1$  using SQualTrack, without the refinement of the measurements space.

### 4.3 Fault Detection as a Constraint Satisfaction Problem

Many engineering problems (e.g., parameter and state estimation, robust control design problems) can be formulated in a logical form by means of some kind of first order predicate formulas: formulas with the logical quantifiers (universal and existential), a set of real continuous functions (equalities and inequalities), and variables ranging over real interval domains.

As defined in [Shary \(2002\)](#), a numerical constraint satisfaction problem is a triple  $\mathcal{CSP} = (\mathcal{V}, \mathcal{D}, \mathcal{C}(x))$  defined by

1. a set of numeric variables  $\mathcal{V} = \{x_1, \dots, x_n\}$ ,
2. a set of domains  $\mathcal{D} = \{D_1, \dots, D_n\}$  where  $D_i$ , a set of numeric values, is the domain associated with the variable  $x_i$ ,
3. a set of constraints  $\mathcal{C}(\mathbf{x}) = \{C_1(\mathbf{x}), \dots, C_m(\mathbf{x})\}$  where a constraint  $C_i(\mathbf{x})$

### 4.3 Fault Detection as a Constraint Satisfaction Problem

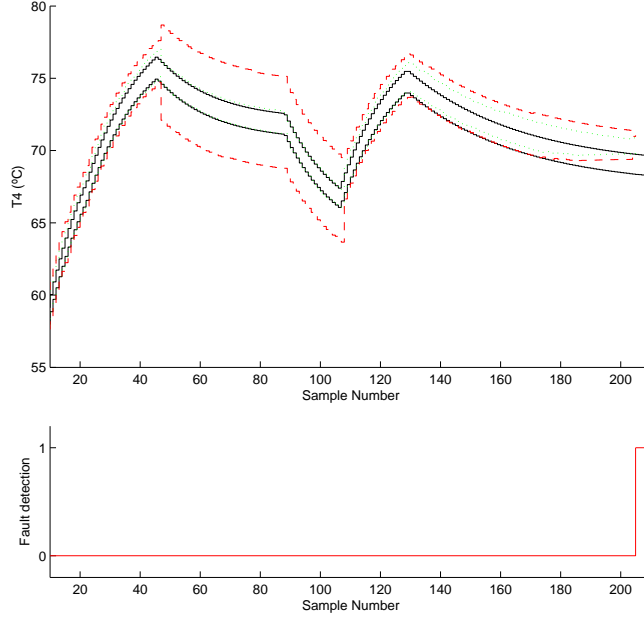


Figure 4.7: A fault in pump  $B1$  using SQualTrack, with the refinement of the measurements space.

is determined by a numeric relation (equation, inequality, inclusion, etc.) linking a set of variables under consideration.

The fault detection problem can be represented by a continuous CSP similar to the one presented in [Jaulin \(2002\)](#), which deals with the problem of nonlinear state estimation. In other words, the task is to solve for the values of the state variables,  $\mathbf{x}(k)$ ,  $1 \leq k \leq n$ , given  $\mathbf{x}(0)$ , input  $\mathbf{u}(k)$ ,  $1 \leq k \leq n$ , and output  $\mathbf{y}(k)$ ,  $1 \leq k \leq n$ ,

Considering that the system dynamics can be modeled in the discrete-time nonlinear form:

$$\begin{cases} \mathbf{x}(k+1) &= \mathbf{g}(\mathbf{x}(k), \mathbf{u}(k), \boldsymbol{\theta}) + \mathbf{w}(k) \\ \mathbf{y}(k) &= \mathbf{h}(\mathbf{x}(k), \mathbf{u}(k), \boldsymbol{\theta}) + \mathbf{v}(k) \end{cases}, \quad (4.16)$$

where:

◇  $\mathbf{u}(k) \in \mathfrak{R}^{n_u}$ ,  $\mathbf{y}(k) \in \mathfrak{R}^{n_y}$ , and  $\mathbf{x}(k) \in \mathfrak{R}^{n_x}$  are the input, output, and state vector, respectively.

### 4.3 Fault Detection as a Constraint Satisfaction Problem

---

- ◇  $\mathbf{w}(k) \in \mathfrak{R}^{n_w}$  and  $\mathbf{v}(k) \in \mathfrak{R}^{n_y}$  are the perturbation and measurement noise vectors, which are unknown but bounded. The perturbation vector takes into account, for instance, unmodeled dynamics of the actual plant, unknown inputs, or an error due to the discretization procedure.
- ◇  $\boldsymbol{\theta} \in \mathfrak{R}^{n_p}$  is a vector of interval bounded parameters, where the parameters values can be considered as time variant or invariant, but always within fixed bounds.
- ◇ The non-linear function  $\mathbf{g}$  relates the state at the time step  $k+1$  to the current time step  $k$ , and the non-linear function  $\mathbf{h}$  relates the state  $\mathbf{x}(k)$  to the measurement  $\mathbf{y}(k)$ .

the CSP corresponding to the dynamic system can be represented as:

$$\begin{aligned}
 \mathcal{V} &= \{\boldsymbol{\theta}, \tilde{\mathbf{y}}(1), \dots, \tilde{\mathbf{y}}(k), \hat{\mathbf{x}}(1), \dots, \hat{\mathbf{x}}(k+1), \tilde{\mathbf{u}}(1), \dots, \tilde{\mathbf{u}}(k) \\
 &\quad \mathbf{w}(1), \dots, \mathbf{w}(k), \mathbf{v}(1), \dots, \mathbf{v}(k)\} \\
 \mathcal{D} &= \{\boldsymbol{\Theta}, \tilde{\mathbf{Y}}(1), \dots, \tilde{\mathbf{Y}}(k), \hat{\mathbf{X}}(1), \dots, \hat{\mathbf{X}}(k+1), \tilde{\mathbf{U}}(1), \dots, \tilde{\mathbf{U}}(k) \\
 &\quad \mathbf{W}(1), \dots, \mathbf{W}(k), \mathbf{V}(1), \dots, \mathbf{V}(k)\} \\
 \mathcal{C} &= \{\hat{\mathbf{x}}(2) = \mathbf{g}(\hat{\mathbf{x}}(1), \tilde{\mathbf{u}}(1), \boldsymbol{\theta}, \mathbf{w}(1)) \\
 &\quad \tilde{\mathbf{y}}(1) = \mathbf{h}(\hat{\mathbf{x}}(1), \tilde{\mathbf{u}}(1), \boldsymbol{\theta}) + \mathbf{v}(1) \\
 &\quad \vdots \\
 &\quad \hat{\mathbf{x}}(k+1) = \mathbf{g}(\hat{\mathbf{x}}(k), \tilde{\mathbf{u}}(k), \boldsymbol{\theta}, \mathbf{w}(k)) \\
 &\quad \tilde{\mathbf{y}}(k) = \mathbf{h}(\hat{\mathbf{x}}(k), \tilde{\mathbf{u}}(k), \boldsymbol{\theta}) + \mathbf{v}(k)\}.
 \end{aligned}$$

Note that the CSP problem (contracting the domains for the variables involved) becomes larger with time. Every time step requires an additional vector  $\mathbf{x}$  to be solved, and a number of additional constraints to satisfy. Therefore, the computational complexity of the solution increases with time. As explained for the SQualTrack, an alternative for overcoming this problem is the use of a *sliding time window*.

Many techniques can be used to solve the continuous CSP. For fault detection, it is specially important to be focused on complete techniques, i.e. the ones that

find all solutions. As explained in Chapter 2, a sound but not complete estimate may result in false alarms, which are a serious concern to be reduced. A survey that covers the state of the art of techniques for solving continuous CSP can be found in Neumaier (2004). These techniques generally include branch and bound codes to split a problem recursively into subproblems. Box reduction techniques can also be used to shrink the initial range of the variables without losing any feasible point. In the following, the techniques used in this thesis are briefly described.

#### 4.3.1 Interval-based Consistency Techniques

Consistency techniques can be used to contract the domains of the involved variables by removing inconsistent values Benhamou *et al.* (1999); Collavizza *et al.* (1999); Cruz & Barahona (2002). In particular for the fault detection application, they are used to guarantee that there is a fault when there is no solution that can be found for the CSP problem, i.e. the observed behavior and the model are inconsistent. The algorithms that are based on consistency techniques are actually "branch and prune" algorithms, i.e., algorithms that can be defined as an iteration of two steps Collavizza *et al.* (1999):

1. Pruning the search space by reducing the intervals associated with the variables until a given consistency property is satisfied.
2. Generating subproblems by splitting the domains of a variable

These techniques can be applied to nonlinear dynamic systems, and their results are not sensitive to strong nonlinearities or nondifferentiabilities in the dynamic system Jaulin (2002).

Most interval constraint solvers are based on either Hull-consistency (also called 2B-consistency) or Box-consistency, or a variation of them Benhamou *et al.* (1999). Hull-consistency and Box-consistency approximate to arc-consistency, widely used in finite domains. Arc-consistency eliminates a value from a variable domain if no compatible value exists in the domain of another variable sharing the same constraint Cruz & Barahona (2002). In continuous domains such enumeration is no longer possible and both hull and box-consistency assume that

### 4.3 Fault Detection as a Constraint Satisfaction Problem

---

the domains of the variables are represented by intervals, so they simply aim at tightening their outer bounds. Hull-consistency guarantees arc-consistency only at the bounds of the variable domains. Roughly speaking, a constraint  $c$  is hull-consistent if, for any variable  $x$ , there exist values in the domains of all other variables which satisfy  $c$  when  $x$  is fixed to its bounds,  $\underline{x}$  and  $\bar{x}$  [Collavizza et al. \(1999\)](#). Existing algorithms decompose the original constraints into a set of primitive constraints, and the property can be enforced by interval arithmetic operations.

**Definition 4.3.1.** *Hull-consistency* [Collavizza et al. \(1999\)](#). Let  $(\mathcal{V}, \mathcal{C})$  be a CSP and  $c \in \mathcal{C}$  a  $k$ -ary constraint over  $(v_1, \dots, v_k)$ .  $c$  is hull-consistent iff:

$$\forall i, V_i = \text{Hull}\{v_i | \exists v_1 \in V_1, \dots, \exists v_{i-1} \in V_{i-1}, \exists v_{i+1} \in V_{i+1}, \dots, \exists v_k \in V_k \\ \text{such that } c(v_1, \dots, v_{i-1}, v_i, v_{i+1}, \dots, v_k) \text{ holds } \}$$

Box-consistency guarantees the consistency of each bound of the domain of each variable with the intervals of the others [Cruz & Barahona \(2002\)](#). The main advantage of this approach is that it tackles the problem of hull-consistency for variables with many occurrences in a constraint.

**Definition 4.3.2.** *Box-consistency* [Collavizza et al. \(1999\)](#). Let  $(\mathcal{V}, \mathcal{C})$  be a CSP and  $c \in \mathcal{C}$  a  $k$ -ary constraint over the variables  $(v_1, \dots, v_k)$ .  $c$  is Box-consistent if, for all  $v_i$  the following relation hold:

$$c(V_1, \dots, V_{i-1}, [\underline{V}_i, \bar{V}_i^+), V_{i+1}, \dots, V_k) \\ c(V_1, \dots, V_{i-1}, (\underline{V}_i^-, \bar{V}_i], V_{i+1}, \dots, V_k)$$

*Specific notation:* if  $a$  is a constant in  $\mathbb{F}$  (a finite subset of  $\mathfrak{R}$ , e.g. the set of floating-point numbers),  $a^+$  (resp.  $a^-$ ) corresponds to the smallest (resp. largest) number of  $\mathbb{F}$  strictly greater (resp. lower) than  $a$ .

The aforementioned techniques are said to be local: each reduction is applied over one domain with respect to one constraint. Better pruning of the variable domains may be achieved if, complementary to a local property, some global properties are also enforced on the overall constraint set.

Besides the choice of the reduction technique, according to the knowledge of the model, the splitting strategy can be chosen to increment the efficiency of the solver.



### 4.3 Fault Detection as a Constraint Satisfaction Problem

**Example 4.3.1.** As it is presented in [Granvilliers & Benhamou \(2006\)](#), consider the system  $\{x_2 = x_1^2, x_1^2 + x_2^2 = 2\}$ , and the initial domains  $\{[-10, 10], [-10, 10]\}$  whose solutions are  $(1, 1)$  and  $(-1, 1)$ . The first constraint leads to reduce the domain of  $x_2$  to  $[0, 10]$ , because  $x_2$  must be positive. The second constraint allows one to contract the domain of  $x_1$  to  $[-\sqrt{2}, \sqrt{2}]$ , and so on. The result is the box  $[-1.19, 1.19] \times [0.76, 1.42]$  as illustrated in Fig. 4.8. Unfortunately, this process does not compute the enclosure  $[-1, 1] \times [1, 1]$  of the solution set. This is due to the locality problem. The locality problem is due to the strategy for reducing domains, since every constraint projection is processed independently.

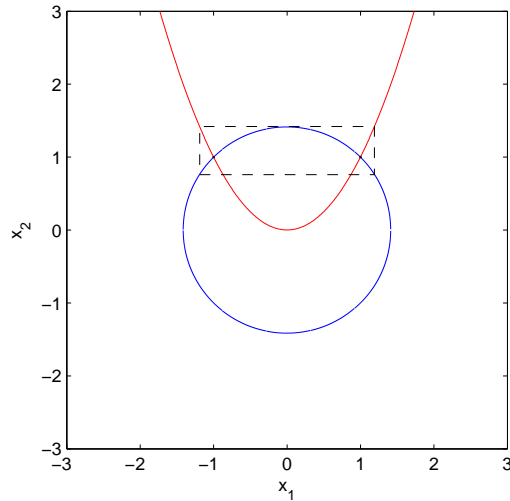


Figure 4.8: Box solution of Example 4.3.1.

In this thesis, the solution of the fault detection *CSP* is performed by using the solver RealPaver [Granvilliers & Benhamou \(2006\)](#).

#### 4.3.1.1 Interval Observer

Instead of using directly the model of a dynamic system given by Eq. 4.16, an observer can be used to correct the estimation of the states by considering a feedback from measured signals.

Taking into account the uncertainty by means of intervals, as defined in [Puig et al. \(2006\)](#), a non-linear interval observer equation with a Luenberger-like structure for a system in the state-space representation can be written as:

$$\begin{aligned}\hat{\mathbf{x}}(k+1) &= \mathbf{g}(\hat{\mathbf{x}}(k), \mathbf{u}(k), \boldsymbol{\theta}) + \mathbf{K}(\mathbf{y}(k) - \hat{\mathbf{y}}(k)), \\ \hat{\mathbf{y}}(k) &= \mathbf{h}(\hat{\mathbf{x}}(k), \mathbf{u}(k), \boldsymbol{\theta}),\end{aligned}\tag{4.17}$$

### 4.3 Fault Detection as a Constraint Satisfaction Problem

---

where  $\hat{\mathbf{x}} \in \mathbb{R}^{nx}$  and  $\hat{\mathbf{y}} \in \mathbb{R}^{ny}$  are estimated state and output vectors of dimension  $nx$  and  $ny$ , respectively,  $\mathbf{u} \in \mathbb{R}^{nu}$  and  $\mathbf{y} \in \mathbb{R}^{ny}$  are measured input and output vectors of dimension  $nu$  and  $ny$ ,  $\boldsymbol{\theta}$  is the vector of uncertain parameters of dimension  $np$  with their values bounded  $\boldsymbol{\theta} \in [\underline{\boldsymbol{\theta}}, \bar{\boldsymbol{\theta}}]$ , and  $\mathbf{K}$  is the gain of the observer. The choice of  $\mathbf{K}$  can be done, for example, by pole placement. The observer behaves like a low-pass filter and thus the pole placement is a compromise between fast fault response and sensitivity to disturbances and noise Nyberg & Nielsen (1998).

The estimated outputs are used to check the consistency of the observations with respect to the system model. Therefore a fault is detected when the measured value is either larger or smaller than the predicted value or in other words, when the output of the model is not consistent with the measured output. This assertion is expressed through the logical statement,

$$(\forall \tilde{\mathbf{y}}(k) \in \tilde{\mathbf{Y}}(k)) (\forall \hat{\mathbf{y}}(k) \in \hat{\mathbf{Y}}(k)) \mathbf{r}(k) \neq \mathbf{0}, \quad (4.18)$$

where  $\mathbf{r}(k) = \tilde{\mathbf{y}}(k) - \hat{\mathbf{y}}(k)$  is a vector of residuals.

The dynamic system (4.17) can be represented as a CSP:

$$\begin{aligned} \mathcal{V} &= \{\boldsymbol{\theta}, \tilde{\mathbf{y}}(1), \dots, \tilde{\mathbf{y}}(k), \hat{\mathbf{y}}(1), \dots, \hat{\mathbf{y}}(k), \hat{\mathbf{x}}(1), \dots, \hat{\mathbf{x}}(k+1), \mathbf{u}(1), \dots, \mathbf{u}(k)\} \\ \mathcal{D} &= \{\boldsymbol{\Theta}, \tilde{\mathbf{Y}}(1), \dots, \tilde{\mathbf{Y}}(k), \hat{\mathbf{Y}}(1), \dots, \hat{\mathbf{Y}}(k), \hat{\mathbf{X}}(1), \dots, \hat{\mathbf{X}}(k+1), \mathbf{U}(1), \dots, \mathbf{U}(k)\} \\ \mathcal{C} &= \{\hat{\mathbf{x}}(2) = \mathbf{g}(\hat{\mathbf{x}}(1), \mathbf{u}(1), \boldsymbol{\theta}) + \mathbf{K}(\tilde{\mathbf{y}}(1) - \hat{\mathbf{y}}(1)) \\ &\quad \hat{\mathbf{y}}(1) = \mathbf{h}(\hat{\mathbf{x}}(1), \mathbf{u}(1), \boldsymbol{\theta}) \\ &\quad \vdots \\ &\quad \hat{\mathbf{x}}(k+1) = \mathbf{g}(\hat{\mathbf{x}}(k), \mathbf{u}(k), \boldsymbol{\theta}) + \mathbf{K}(\tilde{\mathbf{y}}(k) - \hat{\mathbf{y}}(k)) \\ &\quad \hat{\mathbf{y}}(k) = \mathbf{h}(\hat{\mathbf{x}}(k), \mathbf{u}(k), \boldsymbol{\theta})\}. \end{aligned}$$

A problem finding the CSP solution is the continuous increment with time in the computational effort. As it was explained in this chapter, an alternative to overcome this problem is the use of a *sliding time window*.

An application example using real data is presented in Chapter 7. This application deals with the air-intake system of an automotive engine.

## 4.4 Comparing Interval-based Consistency Techniques and Other Methods for Fault Detection

This section presents an empirical comparison of the performance of the two interval-based fault detection techniques used in this thesis, namely SQualTrack and ICTs. Moreover, there is also an empirical comparison of the performance of the ICTs and the performance of one technique based on a statistical decision (in particular an observer implemented as an Extended Kalman Filter and a hypothesis testing scheme based on a Z-test). The last comparison tries to answer a question often asked by experts in this field: which are the main differences between the interval-based techniques and the statistical-based ones?.

### 4.4.1 ICTs vs. SQualTrack

As explained in the previous sections, SQualTrack and ICTs check the consistency between the model and the real process by means of the measurements (Fig. 4.9). One major difference is that, for a time window of length  $\omega$ , SQualTrack makes a prediction  $\omega$ -step ahead of the output of a model, whereas ICTs check the consistency between the model and *all* the measurements within the time window. In this way, using SQualTrack, a fault is detected when the intersection between the external estimate,  $\hat{Y}(k|k - \omega)$ , and the corresponding measurement  $\tilde{Y}(k)$  is void. On the other hand, using ICTs, a fault is detected when no solution is found in the CSP.

Moreover, the computational load of SQualTrack could be very high when the window length exceeds a certain value. As seen in Example 4.2.1, the ARR is generated automatically in a recursive way. Therefore, computing the range of these functions could be a difficult task because of the high number of variables and multi-incidences of variables. In the following example, SQualTrack is used without the two improvements proposed in Section 4.2 due to computational limitations.

## 4.4 Comparing Interval-based Consistency Techniques and Other Methods for Fault Detection

---

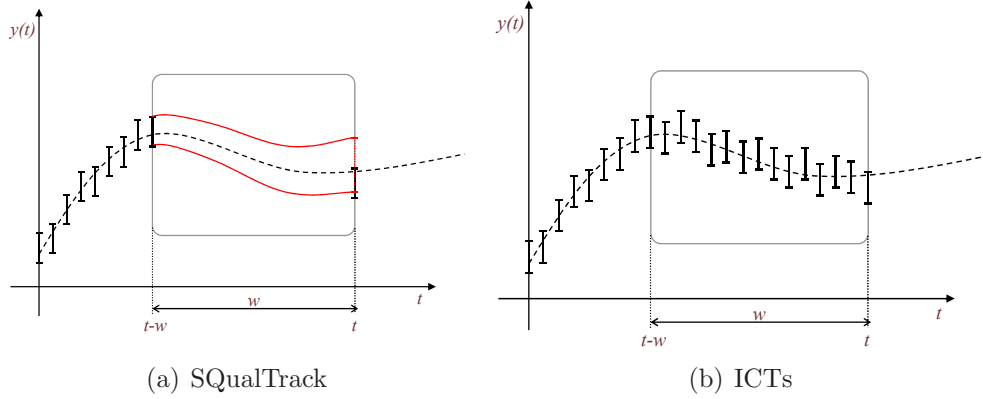


Figure 4.9: Consistency test of the two interval approaches used in this thesis.

### 4.4.1.1 Application Example

A water-tank system is used in this section to illustrate some differences between the performance of the SQualTrack and the ICTs. Figure 4.10 shows a schematic drawing of the system.

The system is composed of two tanks,  $T1$  and  $T2$ , a valve,  $V1$ , and a controller,  $PI1$ , which receives the current level of  $T2$  as the input, and controls a valve,  $V1$ , which regulates the flow of water to  $T1$ .

### Model Equations

The system is described by the elementary analytical relations (EAR) shown in Table 4.2.

The terms  $q_v$ ,  $q_{s1}$  and  $q_{s2}$  denote the volumetric flows,  $x_1$  and  $x_2$  are the heights of the water in tanks  $T1$  and  $T2$ , respectively, and  $u$  is the output signal of the controller. The variables  $u$ ,  $q_v$ ,  $q_{s1}$ ,  $q_{s2}$ ,  $x_1$ , and  $x_2$  are unknown,  $\tilde{u}$ ,  $\tilde{x}_1$ , and  $\tilde{x}_2$  are known variables obtained from sensors, and  $k$ ,  $k_{t1}$ ,  $k_{t2}$ ,  $S_1$ ,  $S_2$ ,  $l_1$  and  $l_2$  are the constant parameters of the system.

Four MSO sets were obtained through the structural analysis, which are minimal with respect to the set of constraints used in the model (Table 4.3). The columns of the table correspond to the elementary analytical relations described

## 4.4 Comparing Interval-based Consistency Techniques and Other Methods for Fault Detection

---

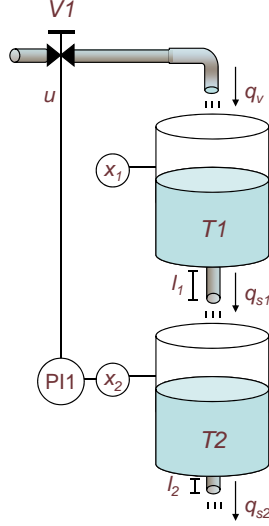


Figure 4.10: Diagram of the coupled water tanks system.

in Table 4.2. The number “1” indicates that the EAR is involved in the corresponding MSO.

For the sake of simplicity, two analytical redundancy relations, obtained from  $MSO_1$  and  $MSO_2$ , are used to compare the approaches. These ARRs are derived from mass balance considerations of tanks  $T1$  and  $T2$ . The corresponding difference equations obtained using the explicit Euler discretisation are:

$$r_1(k) = \tilde{x}_1(k) - \tilde{x}_1(k-1) + \frac{T_s}{S_1} \left( -k \tilde{u}(k-1)^3 + k_{t1} \sqrt{\tilde{x}_1(k-1) + l_1} \right) + w_1(k) \quad (4.19)$$

$$r_2(k) = \tilde{x}_2(k) - \tilde{x}_2(k-1) + \frac{T_s}{S_2} \left( -k_{t1} \sqrt{\tilde{x}_1(k-1) + l_1} + k_{t2} \sqrt{\tilde{x}_2(k-1) + l_2} \right) + w_2(k) \quad (4.20)$$

where  $T_s$  is the sample time, equal to 20 s.

Variables  $w_1$  and  $w_2$  were added to take into account, for example, errors due to the discretization procedure.

All the variables and parameters are considered as intervals for the consistency test using SQualTrack and ICTs. The intervals of the measurements include, for example, the accuracy error and the noise level.

## 4.4 Comparing Interval-based Consistency Techniques and Other Methods for Fault Detection

---

Table 4.2: Elementary analytical relations of the two coupled tanks system

	Elementary Relations	Component
(a)	$q_v = k u^3$	Valve
(b)	$S_1 \frac{dx_1}{dt} = q_v - q_{s1}$	Upper tank
(c)	$q_{s1} = k_{t1} \sqrt{x_1 + l_1}$	Output pipe T1
(d)	$S_2 \frac{dx_2}{dt} = q_{s1} - q_{s2}$	Lower tank
(e)	$q_{s2} = k_{t2} \sqrt{x_2 + l_2}$	Output pipe T2
(f)	$\tilde{u} = u$	D/A converter
(g)	$\tilde{x}_1 = x_1$	$x_1$ sensor
(h)	$\tilde{x}_2 = x_2$	$x_2$ sensor

Table 4.3: *MSO* sets of the two-tank system.

	a	b	c	d	e	f	g	h
$MSO_1$	1	1	1	0	0	1	1	0
$MSO_2$	0	0	1	1	1	0	1	1
$MSO_3$	1	1	0	1	1	1	1	1
$MSO_4$	1	1	1	1	1	1	0	1

Concerning the intervals of the parameters of both ARRs, they were estimated by using strong consistency techniques that guarantee that data from fault free scenarios are covered by the interval model. The identified parameters values are the following ones:  $k \in [0.501, 0.610]10^{-4} \text{ cm}^3.s^{-1}$ ,  $k_{t1} \in [0.839, 0.986] \text{ cm}^{5/2}.s^{-1}$ , and  $k_{t2} \in [1.892, 2.348] \text{ cm}^{5/2}.s^{-1}$ . The values of the other interval parameters are:  $l_1 \in [71.0, 71.5] \text{ cm}$ , and  $l_2 \in [0.0, 0.5] \text{ cm}$ .

### Simulation Results

The faulty scenario that is considered consists of a clogging fault in the output pipe of  $T1$ . The fault occurs at sample 400.

Fault detection results of SQualTrack presented below were obtained using a time window of length 5 samples. Larger window lengths could not be used in this example because of the high computation load and memory consumption. On the other hand, in the case of ICT (using the Weak-3B consistency technique

## 4.4 Comparing Interval-based Consistency Techniques and Other Methods for Fault Detection

---

Granvilliers (2004)), a window length of 50 samples was used.

Figures 4.11 and 4.12 show the results obtained by using the  $ARR_1$ . The fault detection alarm is turned on only by the ICT at sample 408. Figure 4.11 shows that there is not a consistent region of parameters when the fault is detected. Figure 4.12 shows that the interval measurement (solid black line) is within the external estimate (dashed red line) when the fault is present, so the fault could not be detected. The internal estimation is the green dotted line.

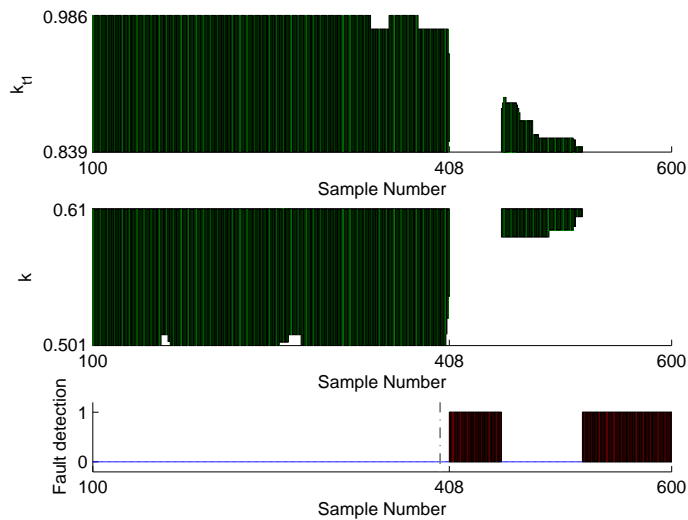


Figure 4.11: Weak-3B Consistency Fault Detection using  $ARR_1$ . The fault is detected from sample 408.

Figures 4.13 and 4.14 show the results obtained by using  $ARR_2$ . The clogging is detected at sample 410 using the ICT.

Notice that for both approaches, when the observed behavior and the model are not proven to be inconsistent, this means that there is not a fault or it could not be detected. In this way, the approaches prioritize to avoid false alarms to missed alarms.

Comparing both approaches, it can be said that ICTs are particularly efficient to check continuous CSPs with large number of variables and constraints. Unlike the SQualTrack, the use of constraint propagation for fault detection, i.e. an iterative reduction of the variable domains, gives an important initial domain reduction because it uses all the measured variables within the time window.

#### 4.4 Comparing Interval-based Consistency Techniques and Other Methods for Fault Detection

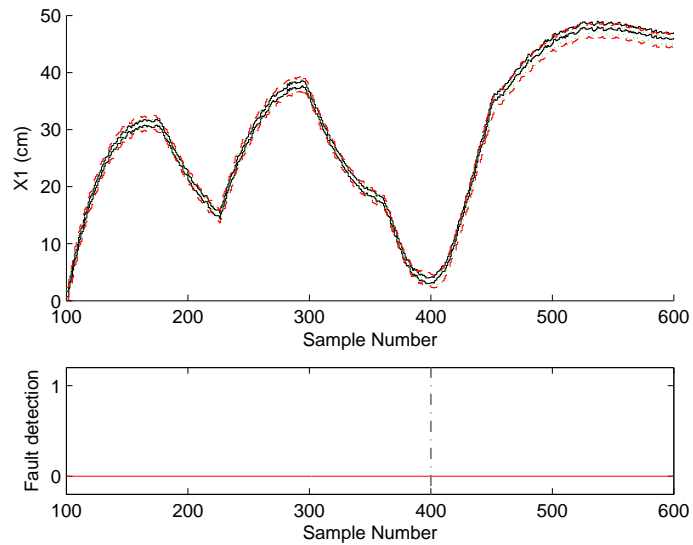


Figure 4.12: SQualTrack Fault Detection using  $ARR_1$ . The fault is not detected.

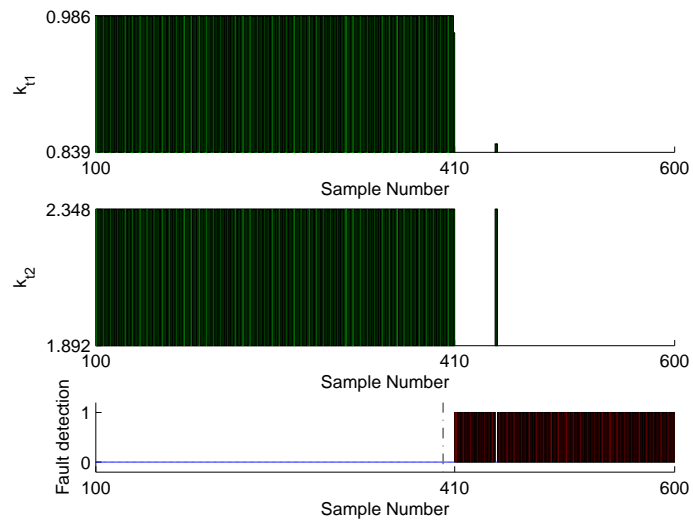


Figure 4.13: Weak-3B Consistency Fault Detection using  $ARR_2$ . The fault is detected from sample 410.

Moreover, by using the ICTs, the fault detection problem is not restricted to ARRs consisting of a single equation as in SQualTrack, since it can be described in a state space representation with multiple state variables.



#### 4.4 Comparing Interval-based Consistency Techniques and Other Methods for Fault Detection

---

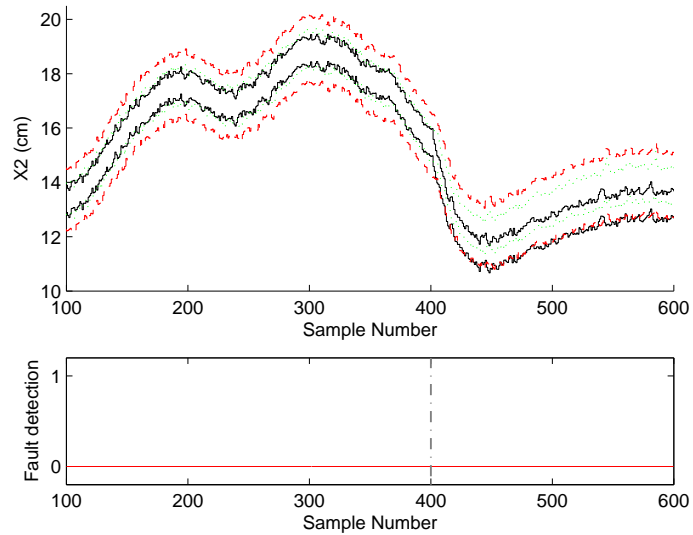


Figure 4.14: SQualTrack Fault Detection using  $ARR_2$ . The fault is not detected.

Finally, SQualTrack provides additional information for fault isolation, as is explained in Chapter 5. The qualitative information that results from comparing the computed output and the measured output, helps to reduce the set of diagnoses.

### 4.4.2 ICTs vs. a Statistical Decision: Extended Kalman Filter and Z-test

In the following, a method for robust model based fault detection is briefly described. This method, applied e.g. in the TRANSCEND system [Biswas \*et al.\* \(2003\)](#); [Mosterman & Biswas \(1999\)](#); [Roychoudhury \*et al.\* \(2006\)](#), includes two modules:

- a robust observer to track the nominal system dynamics,
- a fault detector which monitors the difference between the observed and expected behavior (residual) using a statistical testing method.

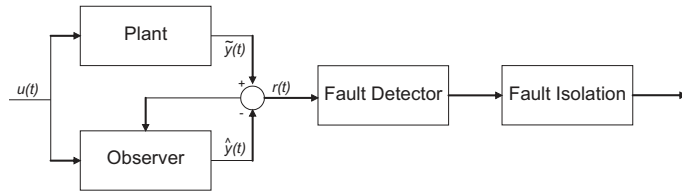


Figure 4.15: Block diagram of the observer-based fault diagnosis approach.

The observer, implemented as an Extended Kalman filter (EKF) [Gelb \(1996\)](#), takes as input the input signals and sensor measurements, and estimates the state  $\mathbf{x}$  as well as outputs, of a discrete-time system that is governed by a non-linear stochastic difference equation such as Eq. 4.16.

The random variables  $\mathbf{w}(k)$  and  $\mathbf{v}(k)$  represent the process modeling errors and measurement noise, respectively. They are assumed to be independent, white and with normal probability distribution  $p(w) \sim N(0, Q)$  and  $p(v) \sim N(0, R)$ , where the process noise covariance  $Q$  and measurement covariance  $R$  matrices are assumed to be constant. The process covariance matrix captures the effect of modeling errors and unknown input disturbances to the system. The measurement covariance matrix models sensor discrepancies and measurement noise.

Because the EKF linearizes estimation around the current time point of the tracking and estimation process, the results obtained are solutions of the linear approximation and are expected to be an approximation of the solution of the

#### 4.4 Comparing Interval-based Consistency Techniques and Other Methods for Fault Detection

---

nonlinear problem. This approximation may not work well in the presence of strong nonlinearities.

The fault detector monitors the measurement residual,  $r(k) = \tilde{y}(k) - \hat{y}(k)$ , at every time step, where  $\tilde{y}$  is the measured value, and  $\hat{y}$  is the expected system output, determined by the observer. Ideally, any non-zero residual value implies a fault, which should trigger the fault isolation scheme. In most real systems, the measured values are corrupted by noise, and the system model (thus the prediction system) is not perfect. Therefore, statistical techniques are required for reliable fault detection.

The fault detector uses a sliding window scheme to compute the residual signal value at time step  $k$ , i.e.,

$$\hat{\mu}_{N_2}(k) = \frac{1}{N_2} \sum_{i=k-N_2+1}^k r(i), \quad (4.21)$$

where  $N_2$  is the predefined window size.

A hypothesis testing scheme based on the Z-test is employed to establish the significance of the deviation. To perform the Z-test, the variance of the measurement residual must be known. The variance of the signal is estimated using Eq. 4.22, but from a larger data sample of size  $N_1$ , i.e.,  $N_1 \gg N_2$  (Fig. 4.16). The considered *VarDelay* guarantees that the variance is computed using data from the system under normal working conditions.

$$\hat{\sigma}_{N_1}^2(k) = \frac{1}{N_1 - 1} \sum_{i=k-N_1+1}^k (r(i) - \mu_{N_1}(k))^2 \quad (4.22)$$

It is assumed that the variance of the signal remains unchanged after fault occurrence.

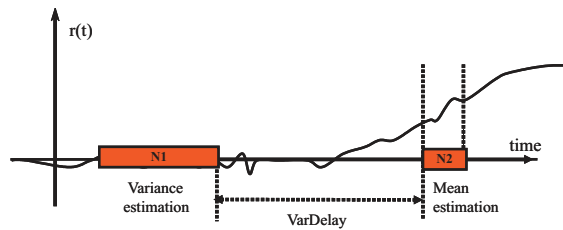


Figure 4.16: Scheme of the fault detection using the Z-test.

## 4.4 Comparing Interval-based Consistency Techniques and Other Methods for Fault Detection

---

The Z-value has distribution  $N(0,1)$ :

$$Z = \frac{\hat{\mu}}{\sigma/\sqrt{N_2}} \quad (4.23)$$

The confidence level, defined by  $\alpha$  (see Fig. 4.17), defines the bound  $[z_-, z_+]$ :

$$P(z_- < z < z_+) = 1 - \alpha \quad (4.24)$$

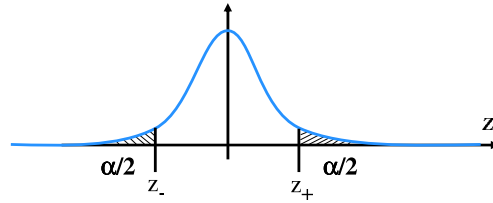


Figure 4.17: Confidence level defined by  $\alpha$ .

For example, for a given confidence level of 95% we get:

$$0.95 = 1 - \alpha = P(z_- < z < z_+) = P(-1.96 < z < 1.96)$$

This bound can be transformed to another bound  $[\mu_-, \mu_+]$  using Eq. 4.23, and the approximation  $\sigma \cong \hat{\sigma}_{N_1}$ :

$$\begin{aligned} \mu_- &= z_- \hat{\sigma}_{N_1} / \sqrt{N_2} \\ \mu_+ &= z_+ \hat{\sigma}_{N_1} / \sqrt{N_2} \end{aligned} \quad (4.25)$$

The Z-test is employed in the following manner:

$$\begin{aligned} \mu_- < \hat{\mu}_{N_2} < \mu_+ &\Rightarrow \text{No Fault} \\ \text{otherwise} &\Rightarrow \text{Fault} \end{aligned}$$

To accommodate measurement noise, inaccuracies in the model and sensitivity of the detection scheme, one has to trade-off false alarm generation versus detection delays. Statistical hypothesis testing schemes help to reduce the false alarm rate, but introduce a delay between the time of occurrence and detection of faults [Roychoudhury et al. \(2006\)](#).

## 4.4 Comparing Interval-based Consistency Techniques and Other Methods for Fault Detection

---

### 4.4.2.1 Summary of Main Differences

The primary differences between the two approaches are presented in Table 4.4.

Table 4.4: Main differences between both techniques.

	<i>EKF and Z-test</i>	<i>Interval-based Consistency Techniques</i>
<i>Uncertainty (hypothesis)</i>	Process and measurement noise are assumed to be <b>Gaussian with zero mean</b>	<b>Unknown but bounded.</b> Intervals take into account, sensor inaccuracies, measurement noise, unmodeled dynamics, errors in the model, uncertainty in the parameters.
<i>Model</i>	The filter may not always converge, because of the linearization approximation	Not sensitive to strong nonlinearities or nondifferentiabilities in the dynamic system.
<i>Fault Detection</i>	Mean value of the residuals within a window is out of a threshold (defined by a confidence level)	When no solution is found in the CSP

Regarding the computational complexity, it can be said that the EKF and Z-test are simpler computationally than the interval-based approach. Moreover, the first approach provides a sign corresponding to the direction of change of the observation that may be useful for the fault isolation task [Mosterman & Biswas \(1999\)](#).

### 4.4.2.2 Application Example

An example of a dynamic system based on two interconnected water tanks is used to compare both techniques. Figure 4.18 shows a schematic drawing of the system, which is composed of two tanks,  $T1$  and  $T2$ , each having an outflow pipe for draining the tanks  $P1$  and  $P2$ , respectively. The first tank also has a source of flow,  $f_{in}$ , for filling the tank.

## 4.4 Comparing Interval-based Consistency Techniques and Other Methods for Fault Detection

---

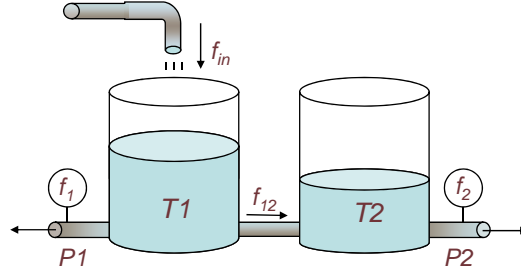


Figure 4.18: The two tank system schematic.

### Model Equations

The discrete time model obtained from the mass balance considerations is composed of the following discrete time equations:

$$\begin{aligned}
 \hat{x}_1(k+1) &= \hat{x}_1(k) + \frac{T_s}{s_1} \left( f_{in}(k) - k_1 \sqrt{\hat{x}_1(k)} - \right. \\
 &\quad \left. - k_{12} \text{sgn}(\hat{x}_1(k) - \hat{x}_2(k)) \sqrt{|\hat{x}_1(k) - \hat{x}_2(k)|} \right) + w_1(k) \\
 \hat{x}_2(k+1) &= \hat{x}_2(k) + \frac{T_s}{s_2} \left( -k_2 \sqrt{\hat{x}_2(k)} + \right. \\
 &\quad \left. + k_{12} \text{sgn}(\hat{x}_1(k) - \hat{x}_2(k)) \sqrt{|\hat{x}_1(k) - \hat{x}_2(k)|} \right) + w_2(k) \\
 \tilde{f}_1(k) &= k_1 \sqrt{\hat{x}_1(k)} + v_1(k) \\
 \tilde{f}_2(k) &= k_2 \sqrt{\hat{x}_2(k)} + v_2(k)
 \end{aligned} \tag{4.26}$$

The terms  $f_{in}$ ,  $f_1$ ,  $f_2$ , and  $f_{12}$  denote the volumetric flows, and  $x_1$  and  $x_2$  are the heights of the water in tanks  $T1$  and  $T2$ , respectively. The variables  $f_{12}$ ,  $x_1$ , and  $x_2$  are unknown, and  $\tilde{f}_1$ , and  $\tilde{f}_2$  are known variables obtained from sensors.  $k_1$ ,  $k_2$ ,  $k_{12}$ ,  $s_1$ , and  $s_2$  are the constant parameters of the system.  $w_i(k)$  is the perturbation vector at time  $k$ , and it takes into account, for example, an error due to the discretization procedure.  $v_i(k)$  is the measurement noise of the interval measurement  $\tilde{f}_i$ . The sample time,  $T_s$ , is equal to 1 second.

For the EKF scheme, a matrix of partial derivatives (the Jacobian) is computed. At each time step the Jacobian is evaluated with current predicted states. This process essentially linearizes the non-linear model around the current estimate. When the height of the tank  $T1$  is greater than the height of the tank  $T2$ ,

#### 4.4 Comparing Interval-based Consistency Techniques and Other Methods for Fault Detection

---

$x_1(k) > x_2(k)$ , the state transition and observation matrices are defined to be the following Jacobians:

$$\mathbf{F}_k = \left. \frac{\partial f}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}_{k-1|k-1}} \quad (4.27)$$

$$= \begin{bmatrix} 1 + \frac{T_s}{s_1} \left( -\frac{k_1}{2\sqrt{x_1}} - \frac{k_{12}}{2\sqrt{x_1-x_2}} \right) & \frac{T_s}{s_1} \frac{k_{12}}{2\sqrt{x_1-x_2}} \\ \frac{T_s}{s_2} \frac{k_{12}}{2\sqrt{x_1-x_2}} & 1 + \frac{T_s}{s_2} \left( -\frac{k_2}{2\sqrt{x_2}} - \frac{k_{12}}{2\sqrt{x_1-x_2}} \right) \end{bmatrix}$$

$$\mathbf{H}_k = \left. \frac{\partial h}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}_k|k-1} = \begin{bmatrix} \frac{k_1}{2\sqrt{x_1}} & 0 \\ 0 & \frac{k_2}{2\sqrt{x_2}} \end{bmatrix} \quad (4.28)$$

Note that for simplicity in the notation the time step subscript  $k$ , in variables  $x_1$  and  $x_2$ , is not used in the previous matrices, but  $x_1$  and  $x_2$  are different at each time step.

The fault detection problem of the two tank system can be represented by a CSP. The set of variables is

$$\begin{aligned} \mathcal{V} = \{ & k_1, k_2, k_{12}, s_1, s_2, \tilde{f}_1(k-\omega), \dots, \tilde{f}_1(k), \tilde{f}_2(k-\omega) \\ & \dots, \tilde{f}_2(k), \hat{x}_1(k-\omega), \dots, \hat{x}_1(k-1), \hat{x}_2(k-\omega) \\ & \dots, \hat{x}_2(k-1), \mathbf{w}(k-\omega), \dots, \mathbf{w}(k-1), \mathbf{v}(k-\omega), \dots \\ & \mathbf{v}(k-1) \} \end{aligned}$$

the set of domains is

$$\begin{aligned} \mathcal{D} = \{ & K_1, K_2, K_{12}, S_1, S_2, \tilde{F}_1(k-\omega), \dots, \tilde{F}_1(k), \tilde{F}_2(k-\omega) \\ & \dots, \tilde{F}_2(k), \hat{X}_1(k-\omega), \dots, \hat{X}_1(k-1), \hat{X}_2(k-\omega) \\ & \dots, \hat{X}_2(k-1), \mathbf{W}(k-\omega), \dots, \mathbf{W}(k-1), \mathbf{V}(k-\omega), \dots \\ & \mathbf{V}(k-1) \} \end{aligned}$$

where

$$\begin{aligned} \hat{X}_1(\cdot) &= [0.00, 0.60] \\ \hat{X}_2(\cdot) &= [0.00, 0.60] \\ W_1(\cdot) &= [-0.001, 0.001] \\ W_2(\cdot) &= [-0.001, 0.001] \end{aligned}$$

## 4.4 Comparing Interval-based Consistency Techniques and Other Methods for Fault Detection

---

The domains  $V_1(\cdot)$  and  $V_2(\cdot)$  considered for different tuning configurations are presented in Table 4.5.

The set of constraints is

$$\begin{aligned}
 \mathcal{C} = \{ & \tilde{f}_1(k-\omega) = k_1 \sqrt{\hat{x}_1(k-\omega)} + v_1(k-\omega) \\
 & \tilde{f}_2(k-\omega) = k_2 \sqrt{\hat{x}_2(k-\omega)} + v_2(k-\omega) \\
 & \hat{x}_1(k-\omega+1) = \hat{x}_1(k-\omega) + \frac{T_s}{s_1} \left( f_{in}(k-\omega) - k_1 \sqrt{\hat{x}_1(k-\omega)} - \right. \\
 & \quad \left. - k_{12} \text{sgn}(\hat{x}_1(k-\omega) - \hat{x}_2(k-\omega)) \sqrt{|\hat{x}_1(k-\omega) - \hat{x}_2(k-\omega)|} \right) + \\
 & \quad + w_1(k-\omega) \\
 & \hat{x}_2(k-\omega+1) = \hat{x}_2(k-\omega) + \frac{T_s}{s_2} \left( -k_2 \sqrt{\hat{x}_2(k-\omega)} + \right. \\
 & \quad \left. + k_{12} \text{sgn}(\hat{x}_1(k-\omega) - \hat{x}_2(k-\omega)) \sqrt{|\hat{x}_1(k-\omega) - \hat{x}_2(k-\omega)|} \right) + \\
 & \quad + w_2(k-\omega) \\
 & \quad \vdots \\
 & \hat{x}_1(k) = \hat{x}_1(k-1) + \frac{T_s}{s_1} \left( f_{in}(k-1) - k_1 \sqrt{\hat{x}_1(k-1)} - \right. \\
 & \quad \left. - k_{12} \text{sgn}(\hat{x}_1(k-1) - \hat{x}_2(k-1)) \sqrt{|\hat{x}_1(k-1) - \hat{x}_2(k-1)|} \right) + \\
 & \quad + w_1(k-1) \\
 & \hat{x}_2(k) = \hat{x}_2(k-1) + \frac{T_s}{s_2} \left( -k_2 \sqrt{\hat{x}_2(k-1)} + \right. \\
 & \quad \left. + k_{12} \text{sgn}(\hat{x}_1(k-1) - \hat{x}_2(k-1)) \sqrt{|\hat{x}_1(k-1) - \hat{x}_2(k-1)|} \right) + \\
 & \quad + w_2(k-1) \\
 & \tilde{f}_1(k) = k_1 \sqrt{\hat{x}_1(k)} + v_1(k) \\
 & \tilde{f}_2(k) = k_2 \sqrt{\hat{x}_2(k)} + v_2(k) \}.
 \end{aligned}$$

### Simulation Results

The scenario considered in this comparison deals with a clogging in the output pipe of the tank  $T2$ . The fault occurs at time 200 s. Two different fault profiles are considered: incipient and abrupt. For the abrupt case, two magnitude deviations are used, a small (5%) change, and a larger change (10%) from the nominal value.

White Gaussian noise (zero mean and standard deviation equal to 1.5% or 3% of the measured signal) was added to the measurements. One hundred runs were



## 4.4 Comparing Interval-based Consistency Techniques and Other Methods for Fault Detection

---

conducted for each noise level, fault size, and tuning parameter combination of both (the statistical decision and the interval-based) approaches.

Table 4.5 lists the parameter values that were used to test the fault detection performance. These parameter values were chosen to show the trade-off between fault detection and false alarm rate. In the EKF, the process noise covariance matrix  $Q$  was varied, and consequently, the gain of the observer, permitting that the actual measurements are “trusted” more and the predicted measurements are trusted less, or the opposite. In the Z-test, the window length  $N_2$  was varied between 5 to 9 samples and the confidence level from  $2.6\sigma$  to  $3\sigma$ . Similarly, in the interval-based technique (using the BC4 consistency technique [Granvilliers \(2004\)](#)), the measurement intervals and the window length were varied.

Table 4.5: Fault Detection system design parameters.

<i>EKF and Z-test</i>	<i>Interval-based Consistency Technique</i>
Window length $N_2 = \{5,7,9\}$	Window length $w = \{5,10,15,20,25\}$
Confidence level $z^+ = \{2.6,2.8,3.0\}$	Measurement Intervals $V_i = \{\pm 2.6\sigma, \pm 2.8\sigma, \pm 3.0\sigma\}$
Window Length $N_1 = 50$	Parameter intervals and perturbation vector $W_i$ between $\pm 0.1\% \dots \pm 1\%$
VarDelay = 100	BC4 consistency technique (local) used
Measurement Covariance $R = \text{constant}$	
Process Noise Covariance $Q = \{0.003, 0.0003, 0.00003\} * I_2$	

Figures 4.19 and 4.20 show the fault detection results for both techniques, for an incipient clogging fault in the output pipe  $P_2$ .

In the statistical decision approach, the tuning parameters for this figure are  $N_2 = 9$ ,  $z^+ = 3.0$ , and  $Q = [0.00003]I_2$ . The vertical lines, in both residuals, indicate the time instant when the fault was detected. In the interval-based approach, the window length is equal to 25, and the interval for the measurement intervals is  $\pm 3.0\sigma$ .

The mean values of the fault detection times (including the standard deviation in parentheses), the false alarm rates, and the missed alarm rates are reported in Tables 4.6 and 4.7. By analyzing the previous information, the following conclusions can be drawn.

## 4.4 Comparing Interval-based Consistency Techniques and Other Methods for Fault Detection

---

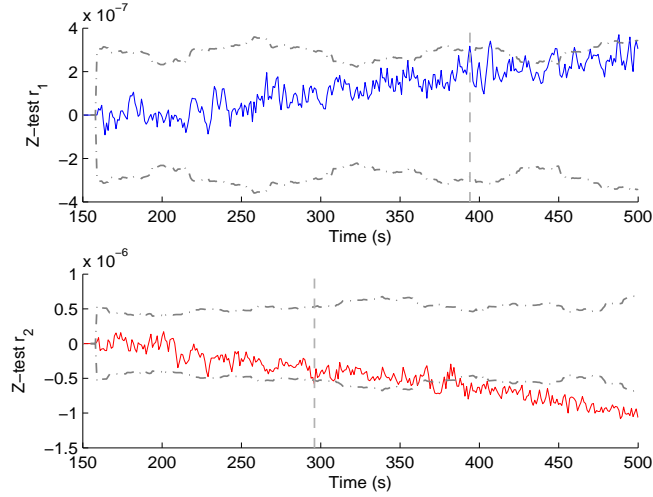


Figure 4.19: EKF and Z-test fault detection.  $N_2 = 9$ ,  $z^+ = 3.0$ , and  $Q = [0.00003]I^2$ .

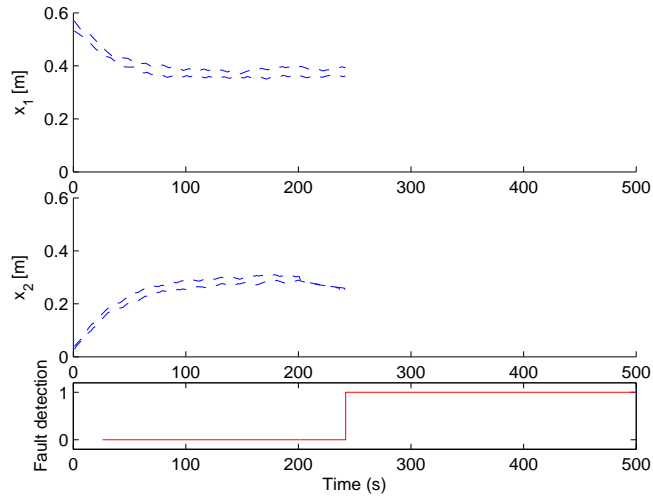


Figure 4.20: Interval-based consistency technique fault detection.  $\omega = 25$ , and  $V_i = \pm 3.0\sigma$ .

Firstly, regarding the interval-based consistency technique:

- When the confidence interval of the noise increases, then False Alarm (FA) rate decreases, and Fault Detection (FD) time or Missed Alarm (MA) rate increases. These relationships can be seen in Fig. 4.21 and 4.22 (incipient fault), and Fig. 4.23 (abrupt fault), where there is a trade-off between FD time/MA rate and the FA rate varying the noise interval.

#### 4.4 Comparing Interval-based Consistency Techniques and Other Methods for Fault Detection

Table 4.6: Results with different fault magnitudes and noise level equal to 1.5% (all times are expressed as time steps from the start of the experiment).

Confidence Interval	EKF and Z-test								
				Incipient Fault		Abrupt Fault 5%		Abrupt Fault 10%	
	N2	Q	FA %	MA %	FD (s)	MA %	FD (s)	MA %	FD (s)
[-2.6 $\sigma$ ,+2.6 $\sigma$ ] (99.07%)	5	1	1	0	404 (33)	6	203 (0,9)	0	202 (0,5)
	7	1	0	0	391 (31)	17	204 (1,7)	0	203 (0,5)
	9	1	0	0	387 (25)	31	205 (2,5)	0	203 (0,4)
	5	2	4	0	378 (40)	5	202 (1,0)	0	202 (0,5)
	7	2	0	0	371 (35)	10	204 (1,7)	0	202 (0,5)
	9	2	0	0	366 (30)	13	205 (2,5)	0	203 (0,4)
	5	3	26	0	245 (24)	0	203 (1,1)	0	202 (0,5)
	7	3	10	0	242 (19)	0	203 (1,8)	0	202 (0,5)
	9	3	1	0	241 (17)	0	204 (2,5)	0	202 (0,2)
[-2.8 $\sigma$ ,+2.8 $\sigma$ ] (99.49%)	5	1	0	0	423 (34)	10	203 (0,8)	0	202 (0,4)
	7	1	0	0	411 (30)	31	204 (1,5)	0	203 (0,5)
	9	1	0	0	399 (26)	51	205 (2,3)	0	203 (0,5)
	5	2	1	0	403 (35)	5	203 (0,9)	0	202 (0,5)
	7	2	0	0	389 (32)	13	204 (1,6)	0	203 (0,5)
	9	2	0	0	383 (26)	27	205 (2,5)	0	203 (0,4)
	5	3	12	0	260 (30)	0	203 (0,9)	0	202 (0,5)
	7	3	4	0	256 (23)	0	203 (1,7)	0	202 (0,5)
	9	3	0	0	251 (19)	0	204 (2,2)	0	203 (0,3)
[-3.0 $\sigma$ ,+3.0 $\sigma$ ] (99.73%)	5	1	0	2	437 (36)	18	204 (0,8)	0	203 (0,3)
	7	1	0	0	427 (29)	44	204 (0,8)	0	203 (0,4)
	9	1	0	0	414 (27)	66	206 (1,0)	0	203 (0,5)
	5	2	0	0	420 (34)	9	204 (0,8)	0	202 (0,4)
	7	2	0	0	408 (29)	29	204 (1,0)	0	203 (0,5)
	9	2	0	0	396 (26)	44	205 (1,3)	0	203 (0,5)
	5	3	5	0	273 (31)	1	203 (0,8)	0	202 (0,4)
	7	3	1	0	269 (28)	0	204 (1,1)	0	202 (0,5)
	9	3	0	0	263 (23)	0	204 (1,4)	0	203 (0,5)

Confidence Interval	Interval-based Consistency Technique							
			Incipient Fault		Abrupt Fault 5%		Abrupt Fault 10%	
	w	FA %	MA %	FD(s)	MA %	FD(s)	MA %	FD(s)
[-2.6 $\sigma$ ,+2.6 $\sigma$ ] (99.07%)	5	28	0	337 (43)	20	201 (55)	0	201 (0,1)
	10	30	0	274 (27)	10	203 (17)	0	201 (0,1)
	15	30	0	252 (19)	4	204 (17)	0	201 (0,1)
	20	30	0	242 (15)	3	204 (16)	0	201 (0,1)
	25	30	0	238 (14)	3	204 (16)	0	201 (0,2)
[-2.8 $\sigma$ ,+2.8 $\sigma$ ] (99.49%)	5	6	0	368 (35)	38	208 (30)	0	201 (0,0)
	10	7	0	291 (23)	23	207 (27)	0	201 (0,0)
	15	7	0	263 (18)	18	207 (26)	0	201 (0,0)
	20	7	0	252 (14)	17	207 (26)	0	201 (0,0)
	25	7	0	246 (13)	17	207 (26)	0	201 (0,1)
[-3.0 $\sigma$ ,+3.0 $\sigma$ ] (99.73%)	5	2	0	391 (34)	64	202 (01)	0	201 (0,0)
	10	2	0	307 (18)	46	202 (01)	0	201 (0,0)
	15	2	0	275 (16)	39	203 (03)	0	201 (0,0)
	20	2	0	260 (14)	39	203 (03)	0	201 (0,0)
	25	2	0	252 (12)	39	203 (03)	0	201 (0,0)

## 4.4 Comparing Interval-based Consistency Techniques and Other Methods for Fault Detection

Table 4.7: Results with different fault magnitudes and noise level equal to 3% (all times are expressed as time steps from the start of the experiment).

Confidence Interval	EKF and Z-test								
				Incipient Fault		Abrupt Fault 5%		Abrupt Fault 10%	
	N2	Q	FA %	MA %	FD (s)	MA %	FD (s)	MA %	FD (s)
[-2.6 $\sigma$ ,+2.6 $\sigma$ ] (99.07%)	5	1	3	71	452 (62)	74	204 (1,2)	5	203 (0,9)
	7	1	0	66	473 (23)	91	206 (1,9)	13	204 (1,7)
	9	1	0	45	474 (26)	96	209 (1,9)	26	205 (2,4)
	5	2	6	11	392 (44)	41	204 (1,1)	1	203 (0,8)
	7	2	1	6	434 (19)	62	205 (1,7)	3	203 (1,0)
	9	2	0	4	431 (33)	79	206 (2,1)	3	204 (1,5)
	5	3	86	0	226 (35)	0	205 (8,1)	0	203 (0,7)
	7	3	67	0	226 (20)	0	207 (1,7)	0	203 (0,8)
	9	3	48	0	226 (20)	0	206 (2,2)	0	203 (0,8)
[-2.8 $\sigma$ ,+2.8 $\sigma$ ] (99.49%)	5	1	0	86	460 (55)	83	204 (1,2)	8	203 (0,9)
	7	1	0	77	476 (50)	94	206 (2,3)	29	204 (1,6)
	9	1	0	70	476 (39)	99	210 (0,0)	47	205 (0,0)
	5	2	4	34	399 (56)	53	203 (1,2)	1	203 (0,9)
	7	2	0	28	445 (41)	79	205 (1,3)	4	204 (1,3)
	9	2	0	10	454 (34)	88	207 (2,6)	5	204 (1,9)
	5	3	68	0	233 (60)	0	207 (4,1)	0	203 (0,7)
	7	3	43	0	229 (32)	2	205 (1,7)	0	203 (0,8)
	9	3	26	0	229 (30)	1	206 (2,3)	0	203 (0,8)
[-3.0 $\sigma$ ,+3.0 $\sigma$ ] (99.73%)	5	1	0	94	458 (14)	90	204 (1,3)	16	204 (1,0)
	7	1	0	88	479 (10)	95	207 (1,7)	41	204 (1,7)
	9	1	0	82	481 (09)	99	210 (0,0)	65	205 (0,0)
	5	2	1	56	426 (14)	68	205 (1,2)	3	203 (0,9)
	7	2	0	51	464 (12)	86	205 (1,6)	6	204 (1,4)
	9	2	0	33	458 (10)	90	208 (2,3)	7	205 (2,0)
	5	3	43	0	237 (16)	1	206 (4,0)	0	203 (0,7)
	7	3	24	0	236 (13)	2	207 (1,9)	0	203 (0,8)
	9	3	15	0	234 (12)	3	206 (2,3)	0	204 (0,8)
Confidence Interval	Interval-based Consistency Technique								
			Incipient Fault		Abrupt Fault 5%		Abrupt Fault 10%		
	w	FA %	MA %	FD(s)	MA %	FD(s)	MA %	FD(s)	
[-2.6 $\sigma$ ,+2.6 $\sigma$ ] (99.07%)	5	40	1	399 (62)	66	323 (97)	15	213 (18)	
	10	43	0	317 (43)	58	307 (95)	0	203 (1,5)	
	15	46	0	285 (33)	52	296 (94)	0	202 (1,1)	
	20	47	0	270 (28)	50	293 (94)	0	201 (1,1)	
	25	47	0	259 (23)	50	293 (94)	0	201 (1,1)	
[-2.8 $\sigma$ ,+2.8 $\sigma$ ] (99.49%)	5	12	17	438 (55)	89	291 (87)	28	211 (18)	
	10	13	0	348 (37)	83	278 (82)	9	204 (1,6)	
	15	14	0	310 (29)	79	274 (81)	1	204 (1,7)	
	20	14	0	288 (25)	74	262 (77)	0	204 (1,8)	
	25	14	0	277 (23)	74	261 (78)	0	204 (1,7)	
[-3.0 $\sigma$ ,+3.0 $\sigma$ ] (99.73%)	5	4	39	456 (46)	96	258 (66)	49	205 (0,9)	
	10	4	0	371 (32)	95	248 (62)	24	204 (1,5)	
	15	4	0	328 (28)	94	242 (57)	11	202 (2,0)	
	20	4	0	304 (22)	93	243 (52)	9	203 (2,5)	
	25	4	0	291 (19)	93	243 (52)	9	203 (2,5)	

## 4.4 Comparing Interval-based Consistency Techniques and Other Methods for Fault Detection

---

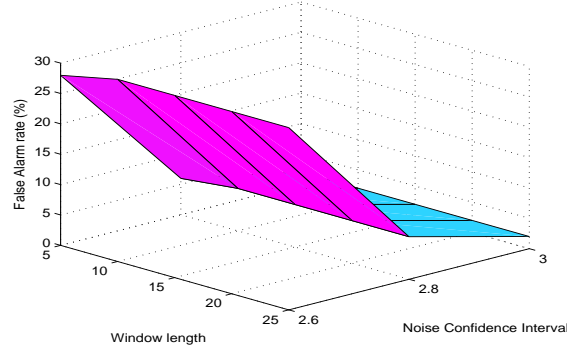


Figure 4.21: False alarm rate using the interval-based technique. Incipient clogging fault in the output pipe  $P2$ .

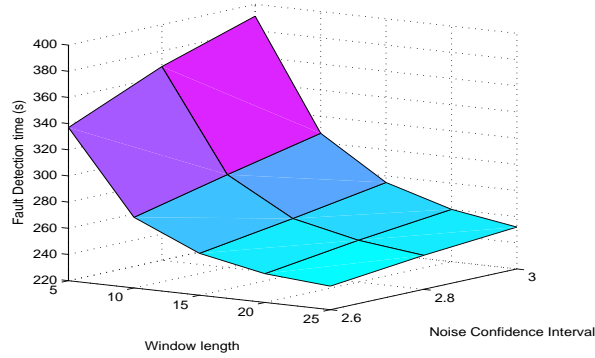


Figure 4.22: Fault detection times using the interval-based technique. Incipient clogging fault in the output pipe  $P2$ .

- For incipient faults, increasing the window length  $\omega$ , decreases the time to FD. In Fig. 4.24, the results indicate that as the noise level in the measurements increases, the FD time also increases.

Secondly, regarding the Extended Kalman Filter and Z-test:

- Similarly to the previous technique, there is a trade-off between FD time/MA rate and the FA rate, when varying the noise confidence interval of the Z-test, as can be seen in Figs. 4.25 and 4.26.
- If the window length  $N_2$  used to compute the mean residual increases, the variations in the computed mean are more smooth, then the FA rate decreases.

## 4.4 Comparing Interval-based Consistency Techniques and Other Methods for Fault Detection

---

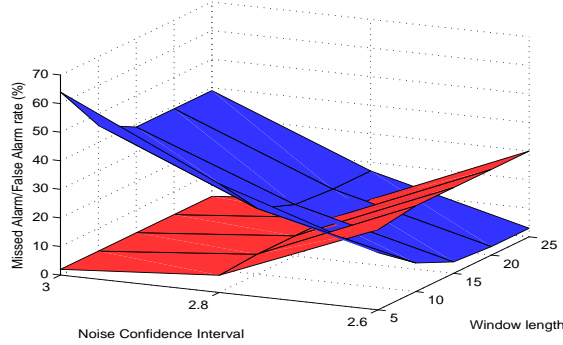


Figure 4.23: Missed detection rate (dark surface) and False alarm rate (light surface) using the interval-based technique. Abrupt clogging fault in the output pipe  $P2$ .

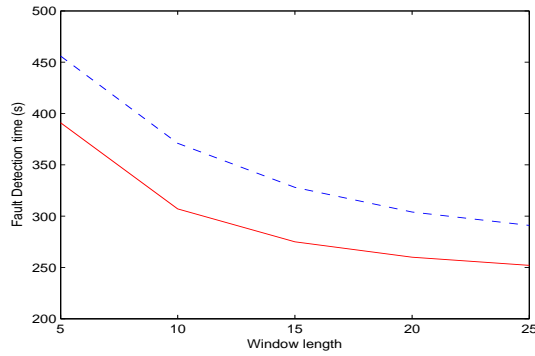


Figure 4.24: Fault detection times with different noise levels using the interval-based technique: 1.5% solid line and 3% dashed line. Incipient clogging fault in the output pipe  $P2$ .

Finally, comparing both approaches, the detection times are similar, and regarding the incipient and the abrupt case:

- Incipient fault. By looking at the results of both methods with similar FA rates, similar FD times can be achieved but with window length  $\omega$  much greater than the window length  $N_2$ . This is because the EKF stores information from the past, whereas for the interval-based technique, the consistency is checked within a window.
- Abrupt fault. EKF and Z-test is more sensitive to smaller magnitudes than the interval based technique, in which the fault is masked by the uncertainty of the model and measurements.

## 4.4 Comparing Interval-based Consistency Techniques and Other Methods for Fault Detection

---

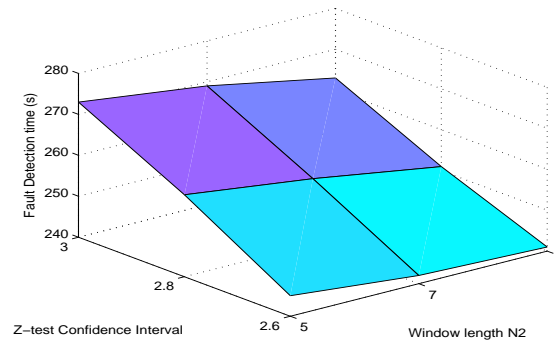


Figure 4.25: Fault detection times using a statistical decision approach. Incipient clogging fault in the output pipe  $P2$ .

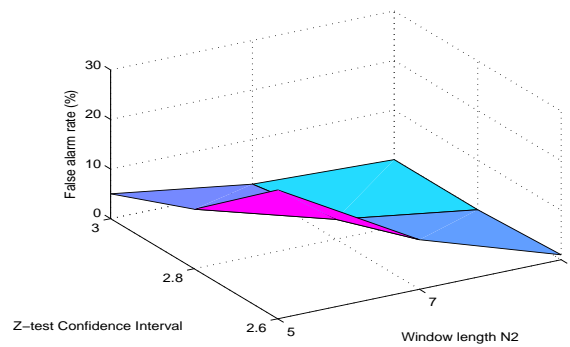


Figure 4.26: False alarm rate using a statistical decision approach. Incipient clogging fault in the output pipe  $P2$ .

## 4.5 Conclusions and Contributions

At the beginning of this chapter, two improvements to the software SQualTrack were proposed to increase the fault detection performance. The dynamic refinement of the parameters space and the measurements space were tested on a simple chemical plant.

Because of some limitations when implementing ARR in SQualTrack, another approach based on interval-based consistency techniques was proposed. The SQualTrack and an ICT were applied to an application example on the same computer. A maximum window length of 5 samples could be used in SQualTrack because of the high computation load and memory consumption. This limitation did not allow it to detect the fault. On the other hand, in the case of an ICT (using the Weak-3B consistency technique), a window length of 50 samples was used and the fault was detected.

Moreover, by using the ICTs, the fault detection problem is not restricted to ARRs consisting of a single equation as in SQualTrack, since it can be described in a state space representation with multiple state variables.

Finally, a technique based on a robust observer and a statistical fault detector was compared to an ICT. In the first approach, the EKF and Z-test are designed for tracking and fault detection in systems where it is reasonable to capture measurement noise and modeling errors as Gaussian processes. In some situations, when the distributions are not Gaussian Orlov (1991) or when the only available information can be expressed as uncertainty bounds (in, e.g., measurements, parameters, and perturbation), interval-based techniques can be used.

Computational complexity of EKF and Z-test technique is lower than in the case of the interval-based technique, in which bisections have to be performed to obtain less overestimation of the solution.

In general, comparing the performance parameters of the fault detection, both techniques yielded similar fault detection results for a proper combination of tuning parameters. In both techniques, a compromise between false alarms, missed detections and detection delays can be made varying the noise confidence interval. Note that the Gaussian noise added in the example, favors the EKF and Z-test technique. In the future, this analysis can be extended to a scenario with bounded



## 4.5 Conclusions and Contributions

---

measurement noise to compare again the performance of both techniques in this situation which favors the interval-based techniques.

# Chapter 5

## Qualitative Fault Diagnosis: Isolation

### Summary

The work presented in this chapter is an attempt to improve the fault isolation in the consistency-based and analytical redundancy approaches. The objective is to reduce adequately the set of diagnoses and to minimize the time required for fault isolation. When enough information becomes available, additional information obtained from the fault models and their effect on the system could be incorporated into the fault isolation task.

In Section 5.3, a diagnosis reasoning in which the signs of the partial derivatives are derived from analytical redundancy relations is proposed. The signs are integrated into the fault signature matrix, to be compared with the deviations of the residuals calculated using the SQualTrack.

Also, this chapter presents a method implemented in the TRANSCEND methodology that generates qualitative fault signatures when a bond graph model of the system is available. This method is based on Temporal Causal Graphs that use the causality of a bond graph and capture causal and temporal relations between parameters and measurements. Afterwards, this technique is briefly compared with the previous one explained in this chapter, and it is extended to the analysis of symbol generation for faults with a discontinuous change in a measurement. Furthermore, a way to distinguish signatures with a discontinuous and non-discontinuous effect is proposed.

The work presented in this chapter was partially developed during my research stay at Vanderbilt University (Nashville-USA), under the supervision of Prof. Gautam Biswas.

## 5.1 Related Work

Some work has been carried out in the topic of taking into account deviations in symptoms (as they are called in the DX community) or in residuals (as they are called in the FDI community).

In [Chang \*et al.\* \(1994\)](#) one approach based on parity equations uses the sensitivity of each parity equation with respect to a fault. The fault isolation stage consists of two steps: to find the degree or level of the fault and to check the consistency of the assumed fault from each parity equation. This approach is restricted to a nominal steady state in a system, so doing a static diagnosis.

In [Olive \*et al.\* \(2003\)](#), two approaches were presented: an interval-based method and a sign-based method. The interval-based method uses an anticipated dictionary of faults and gives bounds in measurements to every fault. In the sign-based method, faults are expressed in terms of deviations w.r.t. the nominal values. By studying the sign of the partial derivatives, the faulty parameters provide a signed influence of the parameters on each test.

The approach proposed in [Console \*et al.\* \(2003\)](#) consists of deriving semi-automatically, qualitative deviation models from quantitative models developed in Matlab<sup>TM</sup>. The general objective is to analyze the behavior in terms of: (i) how a variable deviates from its nominal value when a fault occurs, and (ii) how the deviation of the input variables influence the deviation of the output variables.

In [Puig \*et al.\* \(2005\)](#), a method that combines five fault signature matrices is used for the fault isolation process. The matrices store knowledge about faulty system behavior: boolean fault signal occurrence, signs of residual violation, sensitivities, time of fault signal activation and fault signal occurrence order. This paper is mainly focused in the functions that link each fault signature matrix to events in the residual history.

Another proposal, described in [Calderón \*et al.\* \(2005\)](#), consists of creating a fault library containing qualitative information. Data are obtained from devia-

---

## 5.2 Additional Information Sources for Fault Isolation

tions resulting from interval fault detection. Qualitative information about the deviations is used to create rules for faulty behaviors.

[Kościelny \*et al.\* \(2006\)](#) presents a fault distinguishability study for the DAMADICS benchmark. To improve the fault isolation task, the fault signature matrix is completed with three values,  $\{-1, 0, +1\}$ , using experts' heuristic knowledge about influence of faults on residuals.

The TRANSCEND methodology [Manders \*et al.\* \(2000\)](#); [Mosterman & Biswas \(1999\)](#) employs a qualitative model-based approach for fault isolation. System models are constructed using bond graphs. It utilizes a Temporal Causal Graph (TCG) representation, which is derived directly from the bond graph model of the system. The qualitative effect of a fault is propagated to all measurements using the TCG to determine fault signatures for each measurement. This approach is described in more detail in Section 5.4.

## 5.2 Additional Information Sources for Fault Isolation

In general, the fault isolation task uses a binary vector that contains the result of the fault detection tests, and a fault signature matrix linking faults and ARRs. The diagnosis could be improved if additional information is taken into account, as for example the following sources cited in [Puig \*et al.\* \(2004\)](#):

- the sign of the symptom,
- the sensitivity of the symptom with respect to each fault,
- the order of the symptom appearance of a given ARR with respect to the others,
- the persistency of the symptom indicator for each ARR after a fault,
- the time elapsed between fault occurrence and symptom appearance.

Moreover, these information sources can be extracted from (i) analytical partial models of system with faults, (ii) real data acquisition for all faulty states, and

(iii) expert's heuristic knowledge about influence of faults in residuals [Kościelny et al. \(2006\)](#).

Discriminatory ability can also be improved by adding more sensors, however, in many applications, adding extra sensors may not be cost-effective or physically possible [Daigle \(2008\)](#). Sensors may also be faulty, which adds to the complexity of diagnosis.

In particular, the sign of the symptom using the sensitivity of the fault detection tests with respect to each fault is considered in this dissertation. This work proposes to perform this analysis using the internal form of the residuals. This information completes the boolean fault signature matrix that is compared with the results of, for example, the SQualTrack.

## 5.3 Signs of the Symptoms using the Sensitivity Matrix

The elementary analytical relations of a model can include information on how possible faults can influence the process. These faults may be represented as unknown extra inputs acting on the system (additive faults), or as changes in some plant parameters (multiplicative faults) [Gertler \(1998\)](#). Each ARR can be rewritten by including knowledge about the component fault modes into the model. For example, additive faults can represent plant leaks or biases in sensors, and multiplicative faults can represent clogging or deterioration of plant equipment.

This new ARR associated model has a computational form ( $r^{comp}$ ) and an internal form ( $r^{int}$ ). The computational form is based on known variables, and the internal form is based on, for example, faults, known variables, disturbances and modeling errors. The internal form is not computable because the value of the fault is not known, but it does allow to abstract information about the way in which a fault can act.

As in [Chang et al. \(1994\)](#), the sensitivity  $\mathbf{S}$  is an  $m \cdot n$  matrix (where  $m$  is the number of ARRs, and  $n$  is the number of faults) with the entry  $s_{ji}$  in the  $j$ th row and  $i$ th column. The term  $s_{ji}$  can be viewed as being the sensitivity of

### 5.3 Signs of the Symptoms using the Sensitivity Matrix

---

the model associated with the  $j$ th ARR,  $(r_j)$ , with respect to the  $i$ th fault,  $(f_i)$ . Mathematically, this is expressed as

$$s_{ji} = \frac{\partial r_j^{int}}{\partial f_i} \quad (5.1)$$

The function  $s_{ji}$  depends on the process measurements and system parameters. Then, the sign of the sensitivities,  $\text{sgn}(s_{ji})$ , can be inferred for the domain of the parameters and the measured signals (the function  $\text{sgn}(s_{ji})$  takes the value of +1 when  $s_{ji} > 0$ , -1 when  $s_{ji} < 0$ , and 0 when  $s_{ji} = 0$ ). However, in some cases,  $\text{sgn}(s_{ji})$  can change according to the values of the measurements. Therefore, the corresponding cell of the table of  $\text{sgn}(\mathbf{S})$  cannot be completed beforehand without calculating the sensitivity at each moment.

The possible sign of a fault,  $f$ , can be:

- $\pm 1$ , when  $f > 0$  or  $f < 0$ , e.g. the fault is a bias in a sensor which can be positive or negative; or
- +1 (when  $f > 0$ ) or -1 (when  $f < 0$ ), e.g. the fault is a leak in a tank which always has the same sign.

A new fault signature matrix can be constructed by multiplying the sign of the sensitivity by the sign of the corresponding fault. Then the elements can be:  $\{0, +1, -1, \pm 1, \mp 1\}$ .

- 0, if the fault does not affect the ARR
- +1, when a fault  $f > 0$  affects the symptom with a positive sign
- -1, when a fault  $f > 0$  affects the symptom with a negative sign
- $\pm 1$ , when the sign of a fault is  $\pm 1$ . If  $f > 0$ , the fault affects the symptom with a positive sign, and if  $f < 0$ , the fault affects the symptom with a negative sign
- $\mp 1$ , when the sign of a fault is  $\pm 1$ . If  $f > 0$ , the fault affects the symptom with a negative sign, and if  $f < 0$ , the fault affects the symptom with a positive sign

## 5.3 Signs of the Symptoms using the Sensitivity Matrix

The diagnostic procedure will incrementally generate a set of candidates when a new fault is detected, without providing a transient erratic diagnosis. According to [Cordier \*et al.\* \(2004\)](#), the DX approach follows a row view of the fault signature matrix, considering each line separately corresponding to a confirmed ARR, and isolating the ARR before searching for a common explanation. A fault signature matrix with the sign of the symptom helps to discard some diagnostics.

### 5.3.1 Qualitative Fault Isolation Results using SQualTrack

SQualTrack guarantees that a fault exists when the intersection between the interval measurement and the external envelope is void. Therefore, there are two possibilities for analyzing the internal and computational forms of the model associated with a residual generator: either the external envelope is greater than the interval measurement, and the sign of the symptom would be  $-1$ , or the external envelope is smaller than the interval measurement, and the sign would be  $+1$ . This can be seen in Fig. 5.1, where the interval measurement is shown by the solid lines, and the inner and outer envelope are shown by the dotted and dashed lines, respectively.

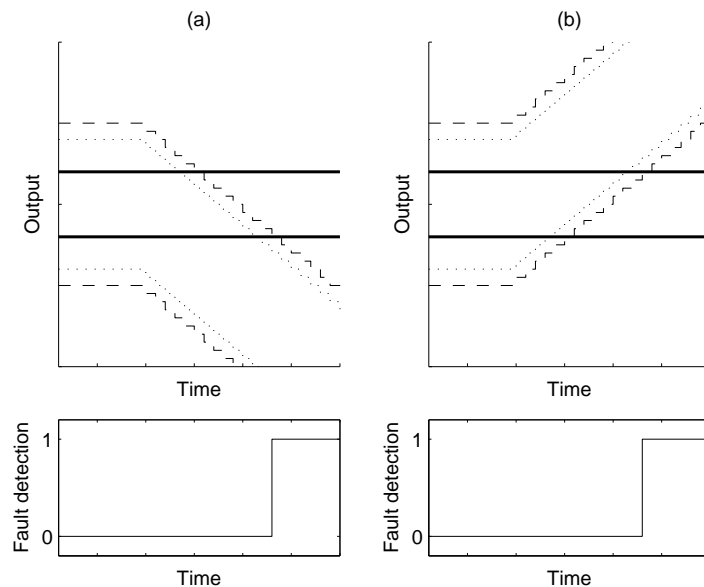


Figure 5.1: SQualTrack Fault Detection: (a) when the sign of the symptom is  $+1$ , and (b) when the sign of the symptom is  $-1$ .

### 5.3.2 An Application to Coupled Water Tanks

A linearized version of the two-tank system introduced in Section 4.4.1 is used to explain the obtainment of additional information from the model for diagnostic reasoning. Linearization is used to simplify the explanation of the proposed strategy for fault diagnosis, but notice that SQualTrack and the proposed method can also be used for nonlinear models.

#### 5.3.2.1 Faults

Assume that nine possible fault scenarios are considered.

- Sensor  $x_1$ :
  - $f_1$ : an additive fault (bias), with  $f_1 > 0$  or  $f_1 < 0$ .
  - $f_2$ : a multiplicative fault, with  $f_2 > 0$ .
- Sensor  $x_2$ :
  - $f_3$ : an additive fault (bias), with  $f_3 > 0$  or  $f_3 < 0$ .
  - $f_4$ : a multiplicative fault, with  $f_4 > 0$ .
- D/A converter:
  - $f_5$ : an additive fault (bias), with  $f_5 > 0$  or  $f_5 < 0$ .
- Tank  $T1$  and its output pipe:
  - $f_6$ : a constant leak, with  $f_6 > 0$ .
  - $f_7$ : a clogging fault in the output pipe, with  $f_7 > 0$ .
- Tank  $T2$  and its output pipe:
  - $f_8$ : a constant leak in tank  $T2$ , with  $f_8 > 0$ .
  - $f_9$ : a clogging fault in the output pipe, with  $f_9 > 0$ .

Extending the model to include faults provides the relationships shown in Table 5.1.



### 5.3 Signs of the Symptoms using the Sensitivity Matrix

---

Table 5.1: Extended model including faults for the two coupled tanks system

	Elementary Relationships
(a)	$q_v = k u$
(b)	$S \dot{x}_1 = q_v - q_{s1} - f_6$
(c)	$q_{s1} = k_{s1} x_1 (1 - f_7)$
(d)	$S \dot{x}_2 = q_{s1} - q_{s2} - f_8$
(e)	$q_{s2} = k_{s2} x_2 (1 - f_9)$
(f)	$\tilde{u} = u + f_5$
(g)	$\tilde{x}_1 = x_1 (1 - f_2) + f_1$
(h)	$\tilde{x}_2 = x_2 (1 - f_4) + f_3$

#### 5.3.2.2 FSM taking into account Residual Signs

The computational forms of the two residual generators analyzed in Section 4.4.1 are

$$r_1^{comp} = S \dot{\tilde{x}}_1 - k \tilde{u} + k_{s1} \tilde{x}_1 \quad (5.2)$$

$$r_2^{comp} = S \dot{\tilde{x}}_2 - k_{s1} \tilde{x}_1 + k_{s2} \tilde{x}_2 \quad (5.3)$$

and the corresponding internal forms are:

$$\begin{aligned} r_1^{int} = & +f_1 k_{s1} - f_2 k \tilde{u} - f_5 k - f_6 + f_7 k_{s1} \tilde{x}_1 + \\ & + f_2 f_6 + f_2 f_5 k - f_1 f_7 k_{s1} \end{aligned} \quad (5.4)$$

$$\begin{aligned} r_2^{int} = & -f_1 k_{s1} + f_2 \left( S \dot{\tilde{x}}_2 + k_{s2} \tilde{x}_2 \right) + f_3 k_{s2} - \\ & - f_4 k_{s1} \tilde{x}_1 - f_7 k_{s1} \tilde{x}_1 - f_8 + f_9 k_{s2} \tilde{x}_2 + \\ & + f_2 f_8 + f_4 f_8 - f_2 f_4 f_8 + f_1 f_4 k_{s1} + \\ & + f_1 f_7 k_{s1} - f_1 f_4 f_7 k_{s1} - f_2 f_3 k_{s2} - \\ & - f_3 f_9 k_{s2} + f_2 f_3 f_9 k_{s2} + f_4 f_7 k_{s1} \tilde{x}_1 - \\ & - f_2 f_9 k_{s2} \tilde{x}_2 \end{aligned} \quad (5.5)$$

In this work, only single faults are considered,  $f_i f_j = 0, \forall i \neq j$ , and so some

### 5.3 Signs of the Symptoms using the Sensitivity Matrix

---

terms in the internal forms can be simplified.

$$r_1^{int} = +f_1 k_{s1} - f_2 k \tilde{u} - f_5 k - f_6 + f_7 k_{s1} \tilde{x}_1 \quad (5.6)$$

$$r_2^{int} = -f_1 k_{s1} + f_2 \left( S \tilde{x}_2 + k_{s2} \tilde{x}_2 \right) + f_3 k_{s2} - f_4 k_{s1} \tilde{x}_1 - f_7 k_{s1} \tilde{x}_1 - f_8 + f_9 k_{s2} \tilde{x}_2 \quad (5.7)$$

By inspecting the internal forms, the structure of the influence of the faults in the ARRs can be concluded to be as shown in Table 5.2. A number “1” in row  $j$  and column  $i$  of the table denotes that fault  $i$  influences the ARR  $j$  ideally.

Table 5.2: Influence of the faults in ARRs.

	$f_1$	$f_2$	$f_3$	$f_4$	$f_5$	$f_6$	$f_7$	$f_8$	$f_9$
$r_1$	1	1	0	0	1	1	1	0	0
$r_2$	1	1	1	1	0	0	1	1	1

The signs of  $s_{ji}$  are shown in Table 5.3. For example, the sensitivity of the ARR  $r_1$  with respect to the fault  $f_7$  is:

$$s_{1,7} = \frac{\partial r_1^{int}}{\partial f_7} = +k_{s1} \tilde{x}_1 \quad (5.8)$$

Since  $k_{s1} > 0$  and  $\tilde{x}_1 > 0$ , the sign of  $s_{1,7}$  is +1.

Table 5.3: Sign of the sensitivity of a symptom with respect to each fault.

	$f_1$	$f_2$	$f_3$	$f_4$	$f_5$	$f_6$	$f_7$	$f_8$	$f_9$
$r_1$	+1	-1	0	0	-1	-1	+1	0	0
$r_2$	-1	+1	+1	-1	0	0	-1	-1	+1

The sign of the model associated with a residual generator by considering the possible variation in the sign of each fault is analyzed for each case in Table 5.4. For example, for  $r_1$  and  $f_1$ , if  $f_1$  is positive, then the symptom of  $r_1$  will be positive. Consequently, if  $f_1$  is negative, then the symptom of  $r_1$  will be negative. This behavior is reflected in the table using  $\pm 1$ .

### 5.3 Signs of the Symptoms using the Sensitivity Matrix

---

Table 5.4: ARRs and their related fault modes using the sign of the symptom.

	$f_1$	$f_2$	$f_3$	$f_4$	$f_5$	$f_6$	$f_7$	$f_8$	$f_9$
$r_1$	$\pm 1$	$-1$	$0$	$0$	$\mp 1$	$-1$	$+1$	$0$	$0$
$r_2$	$\mp 1$	$+1$	$\pm 1$	$-1$	$0$	$0$	$-1$	$-1$	$+1$

#### 5.3.3 Diagnosis Results

Table 5.5 shows the sets of possible diagnostics obtained using the DX approach when the sign of the residuals is used, and when it is not used. The signs help with discarding some diagnostics, and so the sets are reduced.

Table 5.5: Diagnosis using and not using signs.

Symptoms		Diagnosis	Diagnosis using signs
$\text{sgn}(r_1)$	$\text{sgn}(r_2)$		
$-1$	$-1$		$-$
$-1$	$+1$	$f_1, f_2, f_7$	$f_1, f_2$
$+1$	$-1$		$f_1, f_7$
$+1$	$+1$		$-$
$-1$	$0$	$f_1, f_2, f_5, f_6, f_7$	$f_1, f_2, f_5, f_6$
$+1$	$0$		$f_1, f_5, f_7$
$0$	$-1$	$f_1, f_2, f_3, f_4, f_7, f_8, f_9$	$f_1, f_3, f_4, f_7, f_8$
$0$	$+1$		$f_1, f_2, f_3, f_9$

#### 5.3.4 Simulation Results

A faulty scenario involving a clogging fault in the output pipe of  $T1$ ,  $f_7$ , is considered. The values of the variables are represented by intervals to take into account any associated uncertainty in the measurements. The parameters of the model are also taken as intervals for the same reason.

The discrete forms of the ARRs are used in SQualTrack. Then, the compu-

### 5.3 Signs of the Symptoms using the Sensitivity Matrix

---

tational forms of  $r_1^{comp}$  and  $r_2^{comp}$  are introduced as follows.

$$\tilde{x}_1(k) = \tilde{x}_1(k-1) - \frac{T_s}{S} \left( -k \tilde{u}(k-1) + k_{s1} \tilde{x}_1(k-1) \right) \quad (5.9)$$

$$\tilde{x}_2(k) = \tilde{x}_2(k-1) - \frac{T_s}{S} \left( -k_{s1} \tilde{x}_1(k-1) + k_{s2} \tilde{x}_2(k-1) \right) \quad (5.10)$$

Figures 5.2 and 5.3 show a window from SQualTrack for the models of  $r_1$  and  $r_2$ , respectively.

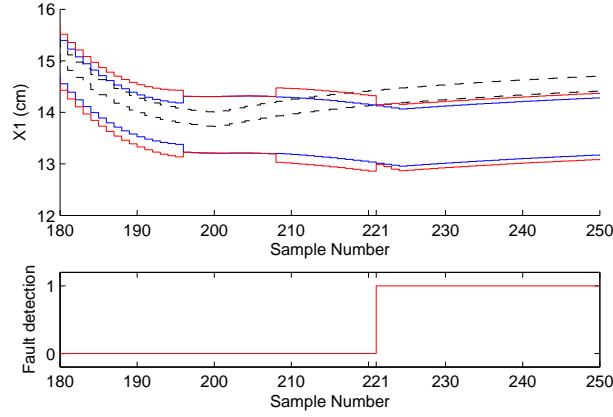


Figure 5.2: SQualTrack Fault Detection using  $r_1$  corresponding to fault  $f_7$  beginning at sample 200. The fault is detected from sample 221.

The upper graphs show the envelopes of the output variable (the inner envelope in dotted lines, and the outer envelope in dashed lines), and the interval measurements in solid lines.

The lower graphs indicate a “1” when a fault is detected.

In all the graphs, the time is expressed in samples, and the sample time  $T_s$  is 10 s. The time windows, of lengths 15 and 25 samples are used. The fault begins at sample 200. For the  $r_2$  case, the fault is detected from sample 214, and for the  $r_1$  case, the fault is detected from sample 221.

Table 5.6 illustrates the diagnostics obtained either by considering or not the signs corresponding at the times in which each symptom appear.

### 5.3 Signs of the Symptoms using the Sensitivity Matrix

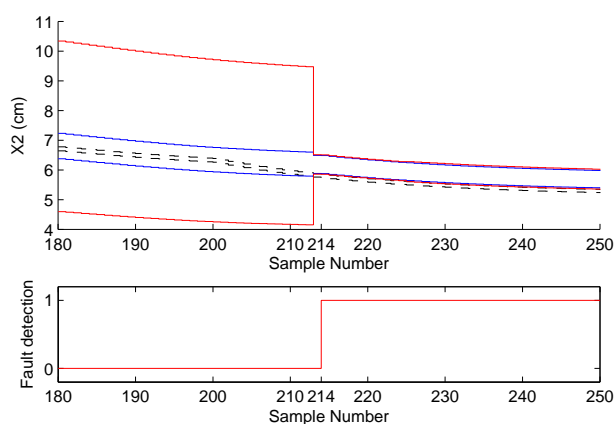


Figure 5.3: SQualTrack Fault Detection using  $r_2$  corresponding to fault  $f_7$  beginning at sample 200. The fault is detected from sample 214.

Table 5.6: Diagnostics for the fault scenario

Sample	214	221
Symptoms	$\text{sgn}(r_1) = 0$ $\text{sgn}(r_2) = -1$	$\text{sgn}(r_1) = +1$ $\text{sgn}(r_2) = -1$
Diagnosis	$f_1 \vee f_2 \vee f_3 \vee f_4 \vee f_7 \vee$ $\vee f_8 \vee f_9$	$f_1 \vee f_2 \vee f_7$
Diagnosis using signs	$f_1 \vee f_3 \vee f_4 \vee f_7 \vee f_8$	$f_1 \vee f_7$

## 5.4 Signs of the Symptoms using Bond Graph Models

Bond Graph (BG) modeling is a method of modeling dynamic systems which was established by Prof. H. Paynter in 1959. BGs are domain-independent, topological, lumped-parameter models that capture the energy exchange mechanisms in physical processes [Karnopp \*et al.\* \(2000\)](#).

The method is based on a central physical concept, energy, and the same methodology is used for different types of systems (such as, e.g. mechanical, electrical, thermic, hydraulic) [Ljung & Glad \(1994\)](#). Another advantage is that the bond graph description of a dynamic system can automatically be translated into state-space equations [Karnopp \*et al.\* \(2000\)](#).

Bond graph is a directed graph whose nodes represent primitive elements, and whose connecting edges, called bonds, represent the transfer of energy between the subsystems.

Each bond is associated with two variables:

- effort, which can correspond to an electrical voltage, a mechanical force, or a hydraulic pressure,
- flow, which can correspond to an electrical current, a mechanical velocity, or a hydraulic volume flow rate.

The product of effort and flow is power, i.e., the rate of energy transfer.

The nodes include [Gawthrop & Bevan \(2007\)](#):

- energy storage:  $C$  and  $I$ .  $C$  component can correspond to an electrical capacitor or a mechanical spring.  $I$  component can correspond to an electrical inductor or a mechanical mass.
- energy dissipation:  $R$ .  $R$  component can correspond to an electrical resistor or a mechanical damper.
- energy transformation:  $TF$  and  $GY$ . The  $TF$  component generalizes an electrical transformer, which has the property that the ratio of voltages (efforts) at the two terminals is the inverse of the ratio of current. The

## 5.4 Signs of the Symptoms using Bond Graph Models

---

*GY* component generalizes a gyrator. The name gyrator arises from the property of a gyroscope that angular velocity (flow) is converted into torque (effort).

- source elements:  $Se$  and  $Sf$ . Source of effort  $Se$  can correspond to an ideal voltage source or an applied force. Source of flow  $Sf$  can correspond to an ideal current source or an applied velocity.

0– and 1–junctions model the equivalent of an electrical parallel (common-effort) and series (common-flow) connection, respectively [Gawthrop & Bevan \(2007\)](#). For a 0–junction, as the Kirchhoffs voltage law, the efforts of all incident bonds are equal, and the sum of flows is zero. Likewise, for a 1–junction, similar to the Kirchhoffs current law, the flows of all incident bonds are equal, and the sum of efforts is zero.

Causality establishes the cause and effect relationships between the factors of power, which are effort and flow. In each bond, the input and output are characterized by the causal stroke which is placed at the end on only one side of a bond.

The assignment of causality to a bond graph can usually be accomplished automatically by computer if the causality is specified at key points on the graph, usually the external ports, and if some general preference for integral or derivative causality is expressed by the modeler [Gawthrop & Bevan \(2007\)](#). After specifying the causality at the external interfaces, it is generally advisable for the modeler to specify the preferred causality of the system  $C$  and  $I$  components. These components may have either integral or derivative causality. For simulation or state-space representations, integral causality is usually preferable since it leads to ordinary differential equations (ODEs), which can be computed without recourse to computationally intensive differential algebraic equation (DAE) solvers.

The causal and temporal relations between system variables can be derived directly from the bond graph model of the system, and represented as a Temporal Causal Graph (TCG) [Mosterman & Biswas \(1999\)](#). It specifies the signal flow graph of the system in a form where edges are labeled with single component parameter values or direct or inverse proportionality relations.

## 5.4 Signs of the Symptoms using Bond Graph Models

---

**Definition 5.4.1.** *Temporal Causal Graph Roychoudhury et al. (2006).* A TCG is a directed graph  $\langle V, L, D \rangle$ .  $V = E \cup F$ , where  $V$  is a set of vertices,  $E$  is a set of effort variables and  $F$  is a set of flow variables in the bond graph system model.  $L$  is the label set  $\{=, 1, -1, p, p^{-1}, p dt, p^{-1} dt\}$  ( $p$  is a parameter name of the physical system model). The  $dt$  specifier indicates a temporal edge relation, which implies that a vertex affects the derivative of its successor vertex across the temporal edge.  $D \subseteq V \times L \times V$  is a set of edges.

The TCG is derived in two steps Mosterman & Biswas (1999):

1. A graph that incorporates cause-effect relations among the power variables in the bond graph is generated.
2. Component parameters and temporal information are added to individual causal edges.

Considering parametric faults, i.e. changes in parameter values of components in the model of a system, the TCG can be used to predict qualitative effects of faults on measurements. In this manner, as the previous method explained in this chapter, this information completes the boolean fault signature matrix to obtain a better fault isolation performance.

The signature is expressed in qualitative terms,  $\{+, 0, -\}$  symbols, of magnitude (zeroth order time-derivative), slope (first order time-derivative) and higher order effects of the residual signal. However, because only the magnitude and slope can be reliably measured, symbol generation only provides us with the immediate magnitude change and the observed slope Daigle (2008).

A forward-propagation algorithm generates the fault signatures Mosterman & Biswas (1999). The qualitative effect of a fault,  $+$  or  $-$ , is propagated to all measurement vertices in the TCG. All deviation propagations start off as zeroth order effects. When an integrating edge is traversed, the magnitude change becomes a first order change. Similarly a first order change propagating across an integrating edge produces a second order change, and so on.



## 5.4 Signs of the Symptoms using Bond Graph Models

**Example 5.4.1.** Consider an open tank with capacity  $C_{t1}$ , and an outflow resistance  $R_1$  for a connected outlet pipe (Fig. 5.4(b)). In Fig. 5.4(a) and 5.4(c), the corresponding BG model and TCG of the system, respectively, are shown.

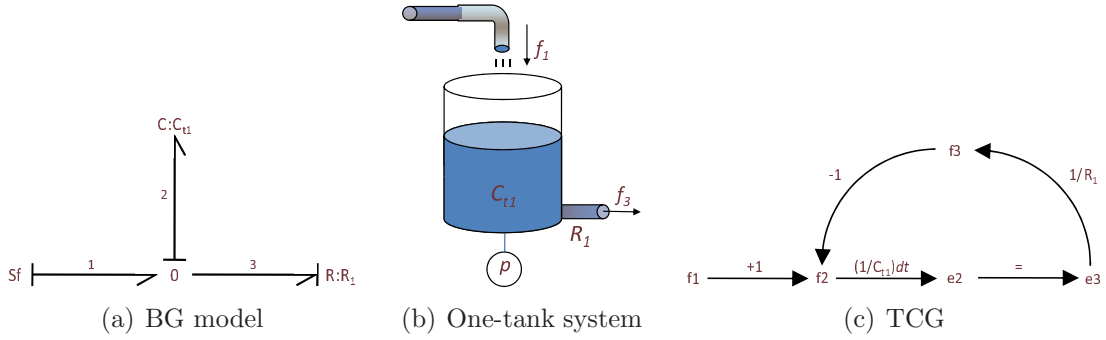


Figure 5.4: One-tank system and its bond graph model and temporal causal graph.

A complete second order signature for effort variable  $e2$  corresponding to fault  $R_1^+$  is  $(0, +, -)$ . The first symbol of the signature represents the immediate direction of change (a discontinuity) at fault occurrence, the second symbol represents the slope of the change after fault occurrence, and the third symbol represents the second order derivative change. Fig. 5.5 shows the forward propagation.  $+$  and  $-$  indicates magnitude changes,  $\uparrow$  and  $\downarrow$  first order change, and  $\uparrow\uparrow$  and  $\downarrow\downarrow$  second order change.

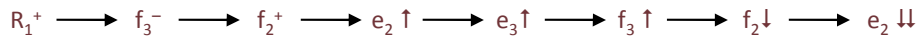


Figure 5.5: Forward propagation for  $R_1^+$  fault.

Fault  $R_1^+$  can be interpreted as a clogging in the output pipe of the tank. It is analogous to fault  $f7$  of the two-tank system analyzed in Section 5.3.2. As expected, both approaches reached similar results. The qualitative effect of fault  $f7$  in  $ARR1$  is  $+1$ , therefore  $\text{sgn}(r_1) = +1 \Rightarrow \tilde{p} > \hat{p}$ . Using this approach an additional information is obtained because the  $(0, +)$  signature implies that at the point of failure, no discontinuous jump in the measurement residual will be observed.

### 5.4.1 Faults with a Discontinuous Change in a Measurement

The set of possible first order fault signatures is given by the combination of symbols  $\{+, 0, -\}$ . Hence, this set is  $\{+-, -+, ++, --, +0, -0, 0+, 0-, 00\}$ . In this thesis, two particular fault signatures,  $\{++, --\}$  are analyzed. As stated in Daigle (2008); Roychoudhury *et al.* (2009), when these fault signatures are related to system parameters, they imply positive feedback, and hence, an unstable system, so are typically disregarded. A special case that may have these signatures corresponds to faults that cause an abrupt change (a discontinuity) in a measurement, as shown below by means of two examples.

**Example 5.4.2.** *Considering again the one-tank system introduced before, firstly, a fault that decreases abruptly the capacity of the tank is analyzed. For example, when something falls into the tank, the capacity  $C$  decreases to  $C^*$ . Since the amount of liquid in the tank is conserved (assuming no overflow), the abrupt change in the capacitance value must reflect as an abrupt change in pressure  $p$  (or equivalently to the height  $h$  of the tank) to  $p^*$ ,  $p^* C^* = p C$ . The capacity  $C^*$  can be expressed as, for example,  $C^* = C(1 - f)$ , where  $f$  is the represents the magnitude of the fault and  $0 < f < 1$ .*

*Forward propagation of this fault,  $C_{t1}^-$ , along the TCG implies*

$$e_2^+ \rightarrow f_3^+ \rightarrow f_2^- \rightarrow e_2?$$

*The first order change of the affected variable  $e_2$  is indeterminate, on the one hand path  $f_2 \rightarrow e_2$  propagates a  $-$  effect, while on the other hand,  $1/C_{t1}$  propagates a  $+$  effect. Qualitatively, it is not possible to predict the first derivative. However, if the tank is emptying or in steady state before the fault (i.e.  $f_2$  is negative or zero), after the fault the first derivative change will be negative. So, in these cases, the signature is  $(+-)$ .*

*Analytically, the slope of the residual associated with  $e_2$ ,  $r = e_2 - \hat{e}_2$ , when the fault occurs is equal to:*

$$\begin{aligned} \text{slope} &= e_2' - \hat{e}_2' = \frac{1}{C_{t1}^*} \left( f_1 - \frac{e_2^*}{R_1} \right) - \frac{1}{C_{t1}} \left( f_1 - \frac{e_2}{R_1} \right) = \\ &= \frac{1}{C_{t1}(1-f)} \left( f_1 - \frac{e_2}{R_1(1-f)} \right) - \frac{1}{C_{t1}} \left( f_1 - \frac{e_2}{R_1} \right) \end{aligned} \quad (5.11)$$

*and for a fix value of  $f_1$ , it could be  $+$ ,  $0$ , or  $-$ , depending on the magnitude of the fault  $f$ , and the value of  $e_2$  when the fault occurs. Figure 5.6 shows the graph of the slope for a one-tank system, function of  $f$  and  $h_2$  (the height of the*

## 5.4 Signs of the Symptoms using Bond Graph Models

tank). Observe that for certain values of  $f$  and  $h_2$ , the slope can be positive (Fig. 5.6(a)), or negative (Fig. 5.6(b)).

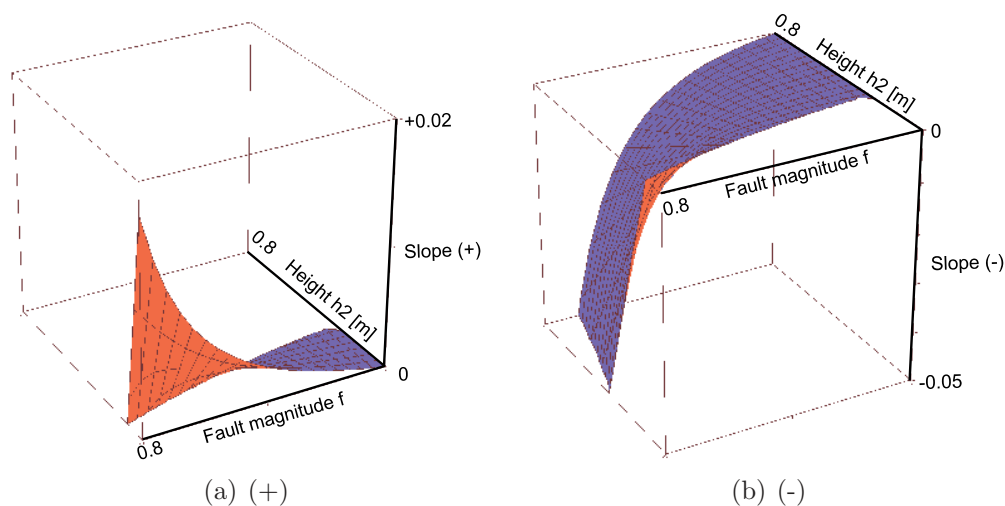


Figure 5.6: Slope of the residual for a one-tank system, function of  $f$  and  $h_2$ .

Figure 5.7 shows the plots of  $h_2$  for the nominal (black solid line) and faulty (red dashed line) behavior when the tank is filling.  $C_{t1}^-$  (20% decrease) is injected at 20 s. A (++) signature is obtained from the residual. Similar plots are shown in Fig. 5.8 and 5.9. In Fig. 5.8, the fault is injected at 50 s and the signature is (+-). In Fig. 5.9, the fault is injected at 20 s when the tank is emptying, and the signature is (+-).

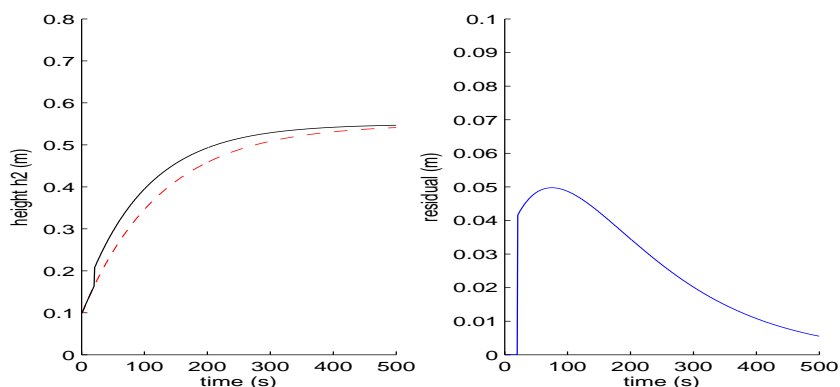


Figure 5.7: Nominal (black solid line) and faulty (red dashed line) behavior for  $h_2$  when the tank is filling.  $C_{t1}^-$  (20% decrease) is injected at 20 s. (++) signature.

## 5.4 Signs of the Symptoms using Bond Graph Models

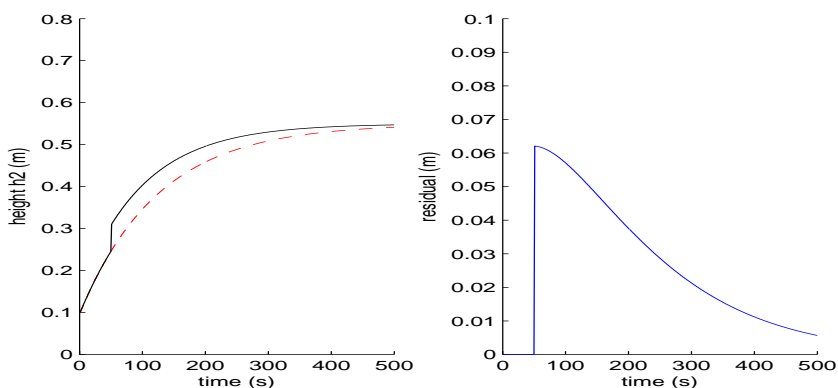


Figure 5.8: Nominal and faulty behavior for  $h_2$  when the tank is filling.  $C_{t1}^-$  (20% decrease) is injected at 50 s. (+-) signature.

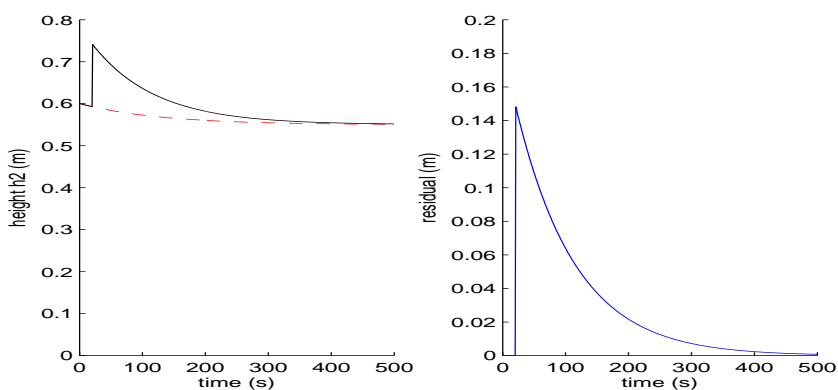


Figure 5.9: Nominal and faulty behavior for  $h_2$  when the tank is emptying.  $C_{t1}^-$  (20% decrease) is injected at 20 s. (+-) signature.

Finally, an abrupt clogging fault in the output pipe of the tank is analyzed. This fault is modeled as an abrupt increase in the pipe resistance and represented as  $R_1^+$ . The effect of this fault in the output flow  $f_3$  makes also a discontinuous jump.

Forward propagation of this fault,  $R_1^+$ , along the TCG implies

$$f_3^- \rightarrow f_2^+ \rightarrow e_2 \uparrow \rightarrow f_3^+$$

The first order change of the affected variable  $f_3$  is indeterminate, on the one hand path  $e_2 \rightarrow f_3$  propagates a + effect, while on the other hand,  $1/R_1$  propagates a - effect. Qualitatively, it is not possible to predict the first derivative.

Figures 5.10 and 5.11 show the plots of  $f_3$  when the tank is filling and  $R_1^+$

## 5.4 Signs of the Symptoms using Bond Graph Models

---

(20% increase) is injected at 20 s and 100 s, respectively. The first figure shows a (--) signature, and the second a (-+) signature.

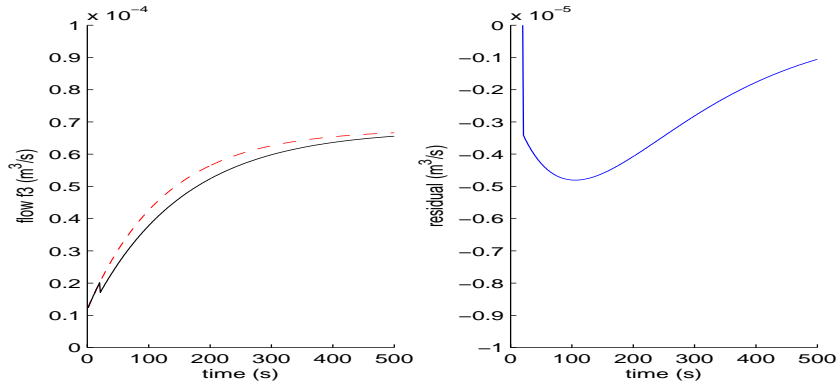


Figure 5.10: Nominal and faulty behavior for  $f_3$  when the tank is filling.  $R_1^+$  (20% increase) is injected at 20 s. (--) signature.

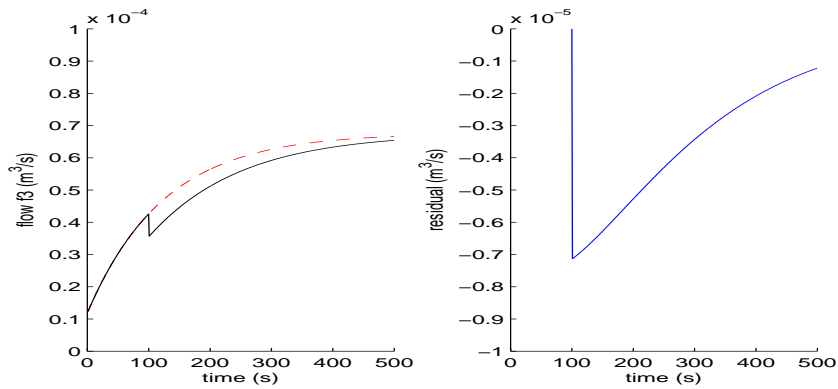


Figure 5.11: Nominal and faulty behavior for  $f_3$  when the tank is filling.  $R_1^+$  (20% increase) is injected at 100 s. (-+) signature.

## 5.4 Signs of the Symptoms using Bond Graph Models

**Example 5.4.3.** A basic mechanical system that consists of a rigid body that can translate is shown in Fig. 5.12(a) Bishop (2002). The system is modeled using a mass, a spring, and a damper. A force,  $F(t)$ , is applied directly to the mass. Figure 5.12(b) illustrates the bond graph model for this system. Two sources are required, one to represent the applied force (effort,  $Se$ ), and a second to represent the fixed based velocity (a flow source,  $Sf$ ). An  $I$  element represents mass, a  $C$  represents the spring, and an  $R$  represents the losses in the damper. In Fig. 5.13, the temporal causal graph of this system is shown.

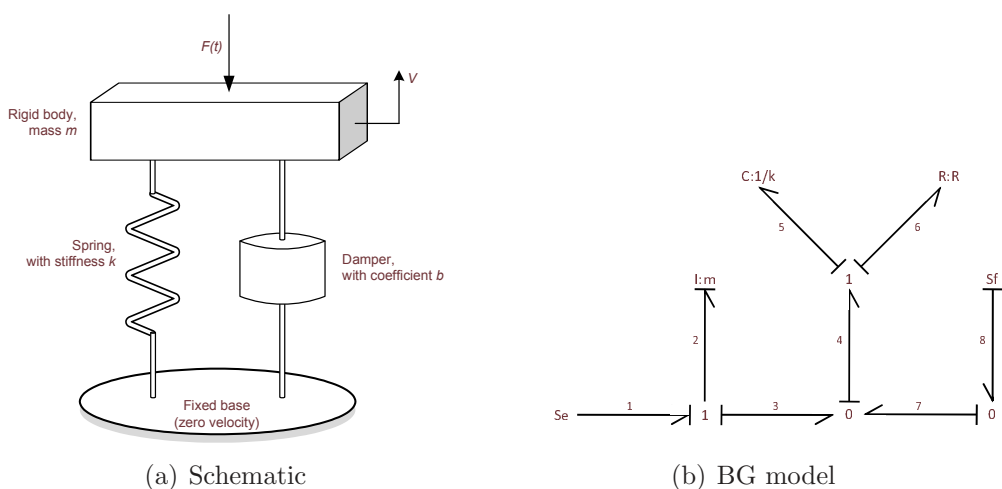


Figure 5.12: Mass-spring-damper system.

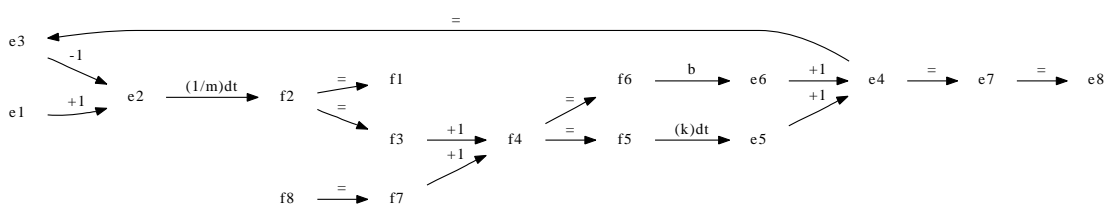


Figure 5.13: Temporal causal graph of the mass-spring-damper system.

The signatures for some faults are shown in Table 5.7. In the system, velocity  $f_2$  is measured. An abrupt fault in the mass causes an abrupt change (a discontinuity) in the measurement of  $f_2$ . As in the previous example, the slope of the residual signal is indeterminate.

Figure 5.14 shows the plots of  $f_2$  when  $m^+$  is injected at 2.2 s, 2.7 s, 3.3 s, and 4.0 s. For this fault, when  $f_2$  is positive, the zeroth order effect is  $-$ , and the

## 5.4 Signs of the Symptoms using Bond Graph Models

---

Table 5.7: Fault signatures when  $f_2$  is measured.

Fault	velocity $f_2$
$b^-$	0+
$b^+$	0-
$k^-$	0+
$k^+$	0-
$m^-$	+?
$m^+$	-?

*opposite. Note that in deriving the signatures using the TCG, it is assumed that variable values are nominally positive.*

### 5.4.2 Symbol Generation using Z-test and Wavelets

When a fault is detected, the zeroth and first order effect of a system measurement can be estimated using, for example, the methodology explained in [Biswas \*et al.\* \(2003\)](#); [Daigle \(2008\)](#). The initial change in a measurement is provided at the time of fault detection for the measurement. If the residual was positive, a + is taken, and if negative, a -. After a deviation is detected in a measurement, then the slope of the residual is calculated using the Z-test. An initial value is computed over a small window starting from the point of fault detection, from which the slope of the residual is determined over another small, but larger window after the end of the smaller window.

The methodology proposed in [Biswas \*et al.\* \(2003\)](#); [Daigle \(2008\)](#) assumes that discontinuities of the form (++) or (--) will not occur (because they imply an unstable system). Then, based on the initial direction of change and the computed slope, it is possible to determine whether a discontinuity has occurred. So, for example, a + in the magnitude and a + in the slope will be reported as (0+), whereas a + in the magnitude and a - (or 0) in the slope will be reported as (+-) (or (+0)).

However, as shown before, signatures (++) or (--) can be reported for some parametric faults in stable systems. Then it is necessary a more advanced symbol generation technique to correctly distinguish (++) from (0+), and (--) from

## 5.4 Signs of the Symptoms using Bond Graph Models

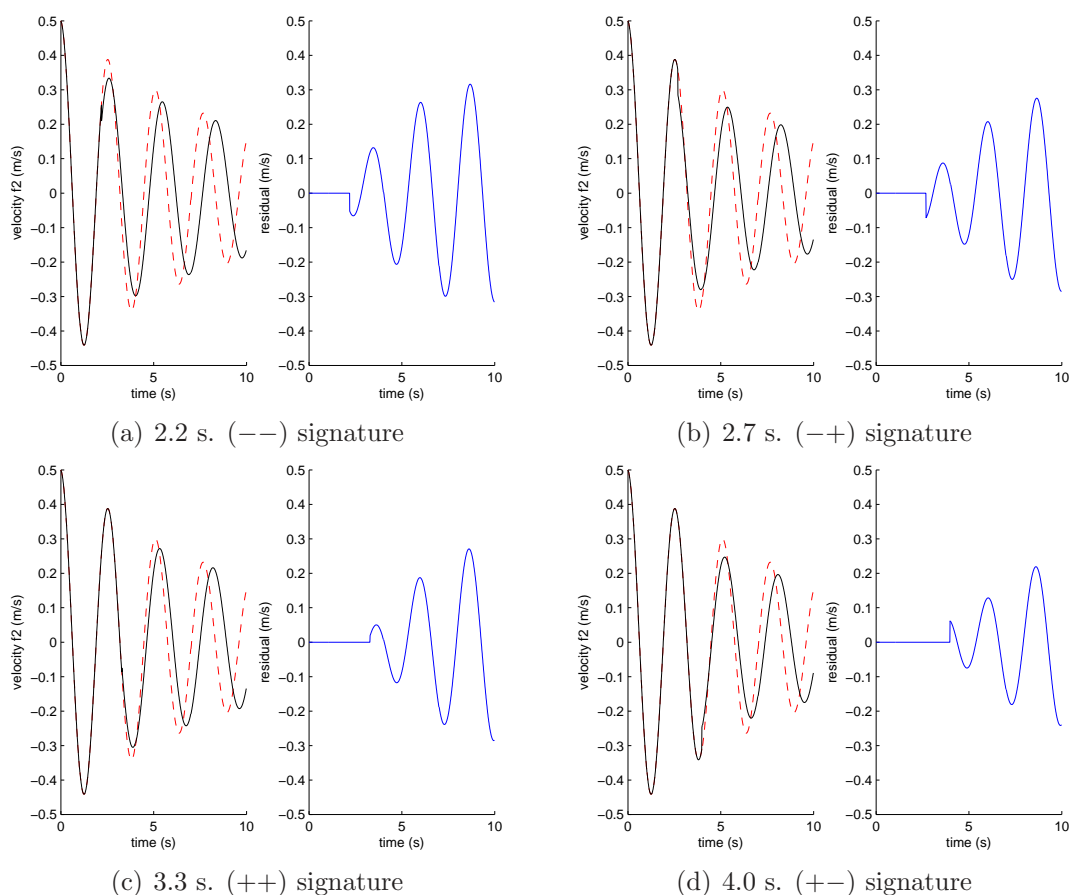


Figure 5.14: Nominal and faulty behavior for  $f_2$  when  $m^+$  is injected at different times.

(0-). One possibility is to use the Wavelet Transform (WT) to detect discontinuous changes in the residual.

WT is a mathematical tool that can decompose a temporal signal into a summation of time-domain basis functions of various frequency resolutions [Chui \(1997\)](#).

In wavelet analysis, two components can be distinguished: the *approximations* (the low-frequency components of the signal), and the *details* (the high-frequency components of the signal). They are recursively computed using digital filters, respectively a lowpass filter and a highpass one. Using the “details” of the discrete wavelet transform (DWT), it is possible to detect the high-frequency components of a discontinuity. Figures [5.15](#) and [5.16](#) show the residuals in a (0+) and (++)



## 5.4 Signs of the Symptoms using Bond Graph Models

case, respectively, with Gaussian noise, when a fault starts at time 3.3 s. DWT (discrete wavelet transform) is applied to these functions, and in particular the Daubechies wavelet [Chui \(1997\)](#) of order 3. The break in [Fig. 5.16](#) is very precisely localized at first-level detail, because the abrupt change contains the high-frequency part and the frequencies in the rest of the signal are not as high.

Note that the DWT can also be used to estimate the fault onset, as was done in [Manders \(2003\)](#).

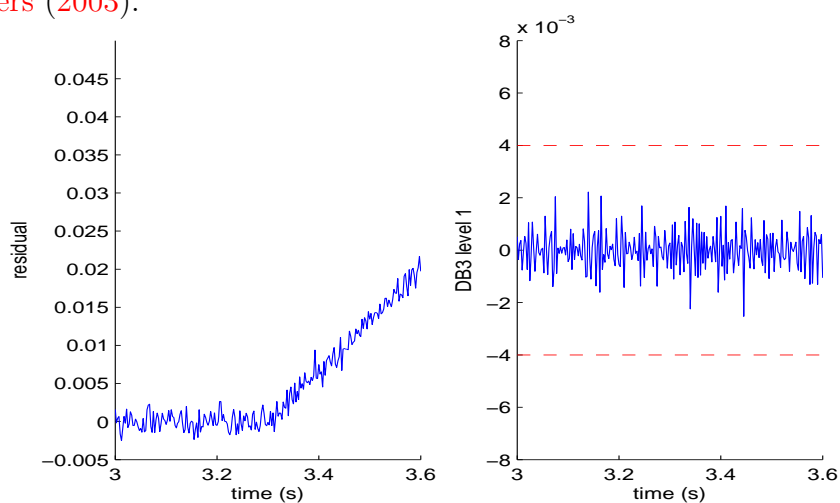


Figure 5.15: DWT applied to a measurement with a non-discontinuous change when a fault occurs.

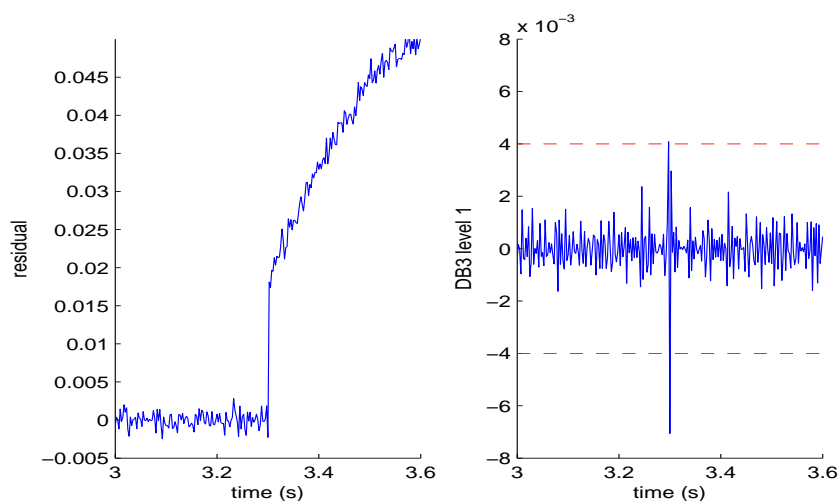


Figure 5.16: DWT applied to a measurement with a discontinuous change when a fault occurs.

## 5.5 Conclusions and Contributions

In this chapter, knowledge about signs obtained from partial derivatives in a quantitative model, is suggested to improve the task of diagnosis. The advantages of an interval tool, the SQualTrack, can be exploited to evaluate the consistency between a model and a system for fault diagnosis.

The residual signs have been analyzed in two ways. First, it is proposed a method that uses the internal form of the consistency relations. This information has been integrated in the fault signature matrix. By comparing the fault signature matrix with the qualitative deviation resulting from the interval detection tool, the set of diagnoses has been reduced.

Second, a method based on bond graph models that derives the qualitative information is analyzed and extended. This extension includes the analysis of faults that cause a discontinuous change in a measurement. In many cases the slope change of these faults is indeterminate because it depends on, for example, the magnitude of the fault. A solution based on wavelets is proposed to distinguish these faults from faults with a non-discontinuous effect.

When comparing the two methods presented in this chapter to derive the qualitative signatures, the advantage of the method based on the TCG is that it can provide more discriminatory evidences than the one based on partial derivatives. It is because fault effects are derived as qualitative magnitude, slope and higher-order effects on individual measurements. On the other side, the advantage of the approach based on partial derivatives is that it provides the sign of the residuals, and it does not require to know if the time dependency of the fault, for example if it is abrupt or incipient, just the sign of the fault.



# Chapter 6

## Quantitative Fault Diagnosis: Isolation and Identification

### Summary

The main contribution of this chapter is the isolation method based on consistency techniques and uncertainty space refinement of interval parameters. The major advantage of this method is that the isolation is fast even taking into account uncertainty in parameters, measurements and model errors. Also, interval calculations bring independency from assumptions of monotony of other authors' previous works, which use observers to solve the isolation problem. Moreover, the method uses the subsystems of the model (ARRs) that are not consistent with the measurements, and the diagnostics obtained by the techniques explained in Chapter 5. In that way the computational load of the parameter estimation routine is lower than using the entire model of the system.

### 6.1 Related Work

Quantitative fault isolation and identification is the final step in the proposed fault diagnosis procedure. The objective of this task is: (i) to refine the fault hypothesis set, because the qualitative fault isolation can not return an unique fault candidate, and (ii), to estimate the magnitude of the fault.

As shown in the previous chapter, qualitative methods for fault isolation are very useful but can not discriminate faults that show no qualitative differences,

or even more, sometimes the qualitative effect is indeterminate. Because of this, parameter estimation techniques can be used to estimate a parameter associated with a fault hypothesis. There is a wide variety of techniques that can be used for this task, e.g. [Milanese \*et al.\* \(1996\)](#); [Simani \*et al.\* \(2003\)](#); [Walter & Pronzato \(1997\)](#).

In [Manders \*et al.\* \(2000\)](#), the fault isolation is based on a qualitative method that uses the temporal causal graph. The isolation is improved by means of a standard least squares parameter estimation method applied to the model in state space form. This model is derived from a bond graph model. A separate parameter estimator is initiated for each fault hypothesis, using the nominal (known) values for all other component parameter values. Fault parameters for which the error term (i.e., the difference between the predicted and observed measurements) do not converge to zero are eliminated from the candidate set. The decision test for convergence to zero is implemented as a statistical hypothesis testing scheme.

A method based in the previous approach is presented in [Samantaray \*et al.\* \(2005\)](#). After an isolation stage, fault parameters that can not be uniquely isolated using a binary fault signature matrix, are estimated in parallel. This method uses ARRs, derived in differential form, from a bond graph model. In this work, the number of ARRs derived equal to the number of sensors installed in the plant. A fault parameter is estimated algebraically by solving an ARR, considering that the rest of the parameters are nominal, and using the mean value of the residual. One limitation of this method is that it uses derivative causality to avoid the estimation of initial conditions when using integral causality. Thus, the estimation of derivatives in noisy environments could be a hard task. Moreover, it needs to solve the parameter associated with a fault hypothesis in a non-linear ARR, which is not always possible.

In [Li & Dahhou \(2007\)](#), a method of fault isolation for non-linear dynamic systems is presented, which assumes that the fault is detected once it occurs and the isolation procedure based on adaptive observers is triggered at this time. That method is based on the monotonous characteristic of an observer prediction error and parameter partitioning.

The approach proposed in this thesis uses the model to predict the outputs, without a correction of the states, when the fault is detected. As a difference to the approach presented in Manders *et al.* (2000), in this thesis a bounded-error estimation approach is proposed, that is only applied to the subsystems of the model that originate discrepancies. Moreover, the approach proposed in this chapter is independent of the assumption required in Li & Dahhou (2007), about the type of nonlinear systems, that the system dynamics is a monotonous function with respect to the considered parameters.

## 6.2 Hypothesis Tests

In this thesis, the quantitative fault isolation and identification methodology uses the framework of structured hypothesis tests proposed in Nyberg (2002). The hypothesis tests used are binary, i.e., the task is to test one null hypothesis against one alternative hypothesis. The null hypothesis and the alternative hypothesis can be written as

$$\begin{aligned}
 H_k^0: & \quad F_p \in M_k \\
 & \quad \text{"some behavioral mode in } M_k \text{ can explain measured data"} \\
 H_k^1: & \quad F_p \in M_k^C \\
 & \quad \text{"no behavioral mode in } M_k \text{ can explain measured data"}
 \end{aligned}$$

Where  $F_p$  denote the present behavioral mode, and  $M_k$  is the set of behavioral modes for the  $k$ th hypothesis test. When  $H_k^0$  is rejected,  $H_k^1$  is assumed to be true, hence, the present behavioral mode can not belong to  $M_k$ . Further, when  $H_k^0$  is not rejected, in most cases nothing is assumed. Each hypothesis test contributes with a piece of information regarding which behavioral modes can be matched to the data or not.

Intervals of the fault parameters that are consistent with the measurements of the system are estimated using interval-based consistency techniques applied to the ARRs in which the fault was detected. The initial domains for all the parameters are substituted by the nominal intervals, except for the parameter or parameters associated with the fault hypothesis. If no solution is found in the CSP of the  $k$ th fault hypothesis, then  $H_k^0$  is rejected and we may drop that corresponding hypothesis.

### 6.2.1 Algorithm for Fault Isolation and Identification

In this thesis the fault isolation/identification problem is also stated as a CSP, similar to the one for fault detection. Considering the  $CSP_{fd}$  (Section 4.3), a general approach for fault isolation can be stated replacing the initial domain of  $\theta$ ,  $\Theta^0$  by  $\Theta^p$ . Where  $\Theta^p$  is the feasible range of parameter variation, given by, e.g. practical considerations. Then, the isolation problem is solved by estimating the consistency parameter range under fault conditions.

A particular case is when at a time, a single independent parameter of the system may be faulty. As a novelty, under this single-fault hypothesis, and considering that:

- $k_{fd}$  is the fault detection time
- $\omega_{max}$  is the maximum sliding time window for fault isolation, and
- $\mathcal{F}$  is the set of fault candidates (possible faulty parameters),

the generic algorithm of the proposed approach is presented in Algorithm 9.

In this algorithm, the fault isolation/identification task starts once the fault has been detected. For each parameter, its initial domain is set to its possible range in practice and the initial domains of the other parameters are equal to the nominal intervals. For example, if we have three candidate parameters  $\theta = (\theta_1, \theta_2, \theta_3)$ , and the corresponding nominal intervals,  $\Theta^0 = (\Theta_1^0, \Theta_2^0, \Theta_3^0)$ , and feasible range of variation in practice,  $\Theta^p = (\Theta_1^p, \Theta_2^p, \Theta_3^p)$ , then three constraint satisfaction problems are solved. For the first, the set of initial domains of the parameters is:  $(\Theta_1^p, \Theta_2^0, \Theta_3^0)$ , for the second,  $(\Theta_1^0, \Theta_2^p, \Theta_3^0)$ , and finally, for the third,  $(\Theta_1^0, \Theta_2^0, \Theta_3^p)$ .

The sliding time window goes up from its smallest value until it gets its maximum possible value. Considering as initial domain the feasible range of variation  $\Theta_i^p$ , when no CSP solution is found, we can decide that the fault is not caused by a value change of the parameter  $\theta_i$ , because no value of  $\theta_i \in \Theta_i^p$  can explain measurement data.

---

**Algorithm 9** Algorithm for Fault Isolation/Identification
 

---

```

1: begin
2:    $\omega = \min(k - k_{fd}, \omega_{max})$ 
3:   for  $i = 1$  to  $n_p$  do
4:      $\Theta = \begin{cases} \Theta_i = \Theta_i^p \\ \Theta_j = \Theta_j^0 \quad \forall j \neq i \in \{1, \dots, n_p\} \end{cases}$ 
5:      $\mathcal{V} = \{ \boldsymbol{\theta}, \tilde{\mathbf{y}}(k-\omega), \dots, \tilde{\mathbf{y}}(k), \hat{\mathbf{x}}(k-\omega), \dots, \hat{\mathbf{x}}(k+1),$ 
6:        $\tilde{\mathbf{u}}(k-\omega), \dots, \tilde{\mathbf{u}}(k), \mathbf{w}(k-\omega), \dots, \mathbf{w}(k),$ 
7:        $\mathbf{v}(k-\omega), \dots, \mathbf{v}(k) \}$ 
8:      $\mathcal{D} = \{ \boldsymbol{\Theta}, \tilde{\mathbf{Y}}(k-\omega), \dots, \tilde{\mathbf{Y}}(k), \hat{\mathbf{X}}(k-\omega), \dots, \hat{\mathbf{X}}(k+1),$ 
9:        $\tilde{\mathbf{U}}(k-\omega), \dots, \tilde{\mathbf{U}}(k), \mathbf{W}(k-\omega), \dots, \mathbf{W}(k),$ 
10:       $\mathbf{V}(k-\omega), \dots, \mathbf{V}(k) \}$ 
11:      $\mathcal{C} = \{ \hat{\mathbf{x}}(k-\omega+1) = \mathbf{g}(\hat{\mathbf{x}}(k-\omega), \tilde{\mathbf{u}}(k-\omega), \boldsymbol{\theta}, \mathbf{w}(k-\omega))$ 
12:        $\tilde{\mathbf{y}}(k-\omega) = \mathbf{h}(\hat{\mathbf{x}}(k-\omega), \tilde{\mathbf{u}}(k-\omega), \boldsymbol{\theta}) + \mathbf{v}(k-\omega)$ 
13:        $\vdots$ 
14:        $\hat{\mathbf{x}}(k+1) = \mathbf{g}(\hat{\mathbf{x}}(k), \tilde{\mathbf{u}}(k), \boldsymbol{\theta}, \mathbf{w}(k))$ 
15:        $\tilde{\mathbf{y}}(k) = \mathbf{h}(\hat{\mathbf{x}}(k), \tilde{\mathbf{u}}(k), \boldsymbol{\theta}) + \mathbf{v}(k) \}$ 
16:      $\mathcal{ESP}_{fi} = (\mathcal{V}, \mathcal{D}, \mathcal{C})$ 
17:      $\Sigma = \text{solution}(\mathcal{ESP}_{fi})$ 
18:     if  $\Sigma = \emptyset$  then
19:       Erase  $\theta_i$  from  $\mathcal{F}$ 
20:     end if
21:   end for
22: end
  
```

---

## 6.3 An Application To Coupled Water Tanks

The non-linear dynamical example of a system based on two coupled water tanks, introduced in Section 4.4.1, is used to explain the quantitative analysis for fault isolation and identification.

The elementary analytical relations of the system are shown again in this chapter, in Table 6.1.

The faults considered, are sensor faults, actuator faults and process faults (leakages and clogging in the output pipes of  $T1$  and  $T2$ ). In this example we assume that only single, abrupt faults occur. In the case of incipient faults, for the fault identification, the methodology explained in this chapter can also be applied. The model of an incipient fault can be represented as in Roychoudhury



### 6.3 An Application To Coupled Water Tanks

Table 6.1: Elementary analytical relations of the two coupled tanks system

	Elementary Relations	Component
(a)	$q_v = k u^3$	Valve
(b)	$S_1 \frac{dx_1}{dt} = q_v - q_{s1}$	Upper tank
(c)	$q_{s1} = k_{t1} \sqrt{x_1 + l_1}$	Output pipe T1
(d)	$S_2 \frac{dx_2}{dt} = q_{s1} - q_{s2}$	Lower tank
(e)	$q_{s2} = k_{t2} \sqrt{x_2 + l_2}$	Output pipe T2
(f)	$\tilde{u} = u$	D/A converter
(g)	$\tilde{x}_1 = x_1$	$x_1$ sensor
(h)	$\tilde{x}_2 = x_2$	$x_2$ sensor

*et al.* (2006), adding a drift term to the nominal parameter value.

The following behavioral modes are considered:

- Gain-fault in the height sensor of  $T1$ . The model corresponding to this behavioral mode is obtained by using Table 6.1, but replacing equation (g) with  $\tilde{x}_1 = x_1(1 - f_1)$ , where  $f_1$  represents a gain fault and is  $0 < f_1 < 1$  when this fault is present.
- Gain-fault in the height sensor of  $T2$ . This behavioral mode is obtained by using Table 6.1, but replacing equation (h) with  $\tilde{x}_2 = x_2(1 - f_2)$ , where  $f_2$  represents a gain fault and is  $0 < f_2 < 1$  when this fault is present.
- Bias in the D/A converter. This behavioral mode is obtained from the fault free model, but replacing equation (f) with  $\tilde{u} = u + f_3$ , where  $f_3$  is a constant bias and  $f_3 \neq 0$  when this fault is present.
- Leakage in  $T1$ . As explained before, but replacing equation (b) with  $S_1 \frac{dx_1}{dt} = q_v - q_{s1} - f_4 \sqrt{x_1}$ . When this fault is present  $f_4 > 0$ .
- Clogging fault in the output pipe of  $T1$ . This behavioral mode is obtained from the fault free model, but replacing equation (c) with  $q_{s1} = k_{t1}(1 - f_5) \sqrt{x_1 + l_1}$ . When this fault is present  $0 < f_5 < 1$ .

## 6.3 An Application To Coupled Water Tanks

---

- Leakage in  $T2$ . This behavioral mode is obtained from the fault free model, but replacing equation (d) with  $S_2 \frac{dx_2}{dt} = q_{s1} - q_{s2} - f_6 \sqrt{x_2}$ . When this fault is present  $f_6 > 0$ .
- Clogging fault in the output pipe of  $T2$ . This behavioral mode is obtained from the fault free model, but replacing equation (e) with  $q_{s2} = k_{t2}(1 - f_7) \sqrt{x_2 + l_2}$ . When this fault is present  $0 < f_7 < 1$ .

Considering the two ARR<sub>s</sub> that correspond to the mass balance of each tank (Eqs. 4.19 and 4.20), the fault signature matrix can be deduced as shown in Table 6.2.

Table 6.2: Fault signature matrix.

	$f_1$	$f_2$	$f_3$	$f_4$	$f_5$	$f_6$	$f_7$
$r1$	1	0	1	1	1	0	0
$r2$	1	1	0	0	1	1	1

### 6.3.1 Simulation Results

To illustrate the performance of the diagnosis system, a faulty scenario involving a clogging fault in the output pipe of  $T1$ ,  $f_5$ , is considered. The values of the measured variables are collected once per 20 seconds, and the sensors of  $x_1$  and  $x_2$  have an uncertainty of  $\pm 0.5$ cm. The parameters of the model are also represented by intervals to take into account any associated uncertainty in the model. Fig. 6.1 shows the measured signals for this scenario. The fault has a magnitude of 10% of  $k_{t1}$ , and begins at sample 225.

A sliding time window of length 20 samples is used for the fault detection. The fault is detected at sample 232 using ARR<sub>1</sub>, and the fault isolation, based on the fault signature (Table 6.2), shows the initial list of four candidates,  $\{f_1, f_3, f_4, f_5\}$ . For the ARR<sub>2</sub> case, the fault is detected at sample 242, and then, two of the initial candidates are dropped,  $\{f_3, f_4\}$ .

### 6.3 An Application To Coupled Water Tanks

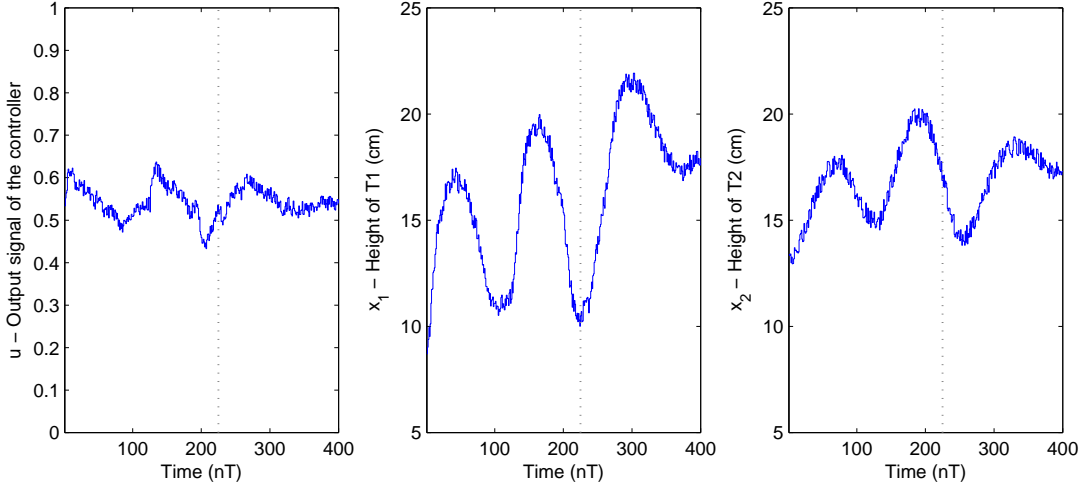


Figure 6.1: Measured signals when a clogging fault in the output pipe of  $T1$  is present.

To show in this scenario the efficiency of the quantitative fault isolation and identification, we did not take into account the signs in the fault signature matrix, which would lead to a faster diagnosis.

The quantitative fault isolation and identification starts for each ARR whose fault detection alarm was activated, for ARR1 it starts at sample 232 and for ARR2 it starts at sample 242. The estimation of parameters  $f_1$  and  $f_5$  using ARR1 and ARR2 is represented in Figs. 6.2 and 6.3. The final estimated interval of parameter  $f_1$  is given by the intersection of the two intervals obtained by using ARR1 (Fig. 6.2a) and ARR2 (Fig. 6.2b), and is depicted in Fig. 6.2c. Similarly, the final estimated interval of parameter  $f_5$  is given by the intersection of the two intervals obtained by using ARR1 (Fig. 6.3a) and ARR2 (Fig. 6.3b), and is depicted in Fig. 6.3c.

Where the intersection of two intervals  $X = [\underline{x}, \bar{x}]$  and  $Y = [\underline{y}, \bar{y}]$  is defined by

$$X \cap Y \triangleq \{z \in \mathfrak{R} \mid z \in X \text{ and } z \in Y\},$$

and satisfies

$$\begin{aligned} X \cap Y &= [\max\{\underline{x}, \underline{y}\}, \min\{\bar{x}, \bar{y}\}] \quad \text{if } \max\{\underline{x}, \underline{y}\} \leq \min\{\bar{x}, \bar{y}\}, \\ &= \emptyset \quad \text{otherwise.} \end{aligned}$$

### 6.3 An Application To Coupled Water Tanks

In the example, at sample 270, the fault hypothesis of a gain-fault in the height sensor of  $T1$  ( $f_1$ ) is rejected after no solution was found in the CSP of the ARR2. At sample 332, the final estimated interval of parameter  $f_5$  is equal to  $[0.0525, 0.1598]$ , and as it was expected, it includes the real value of the fault, 0.1.

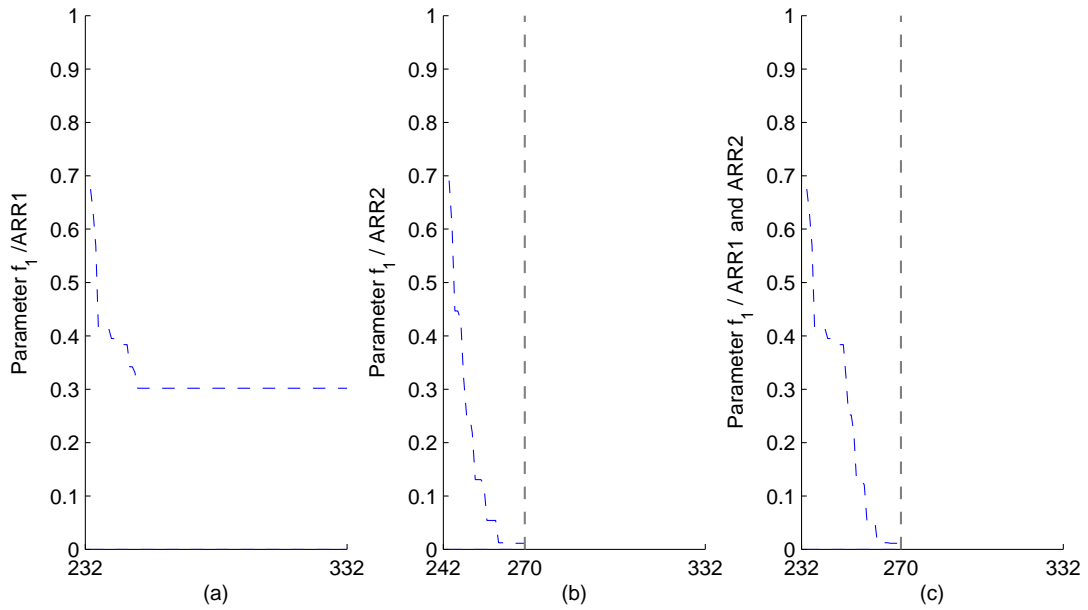
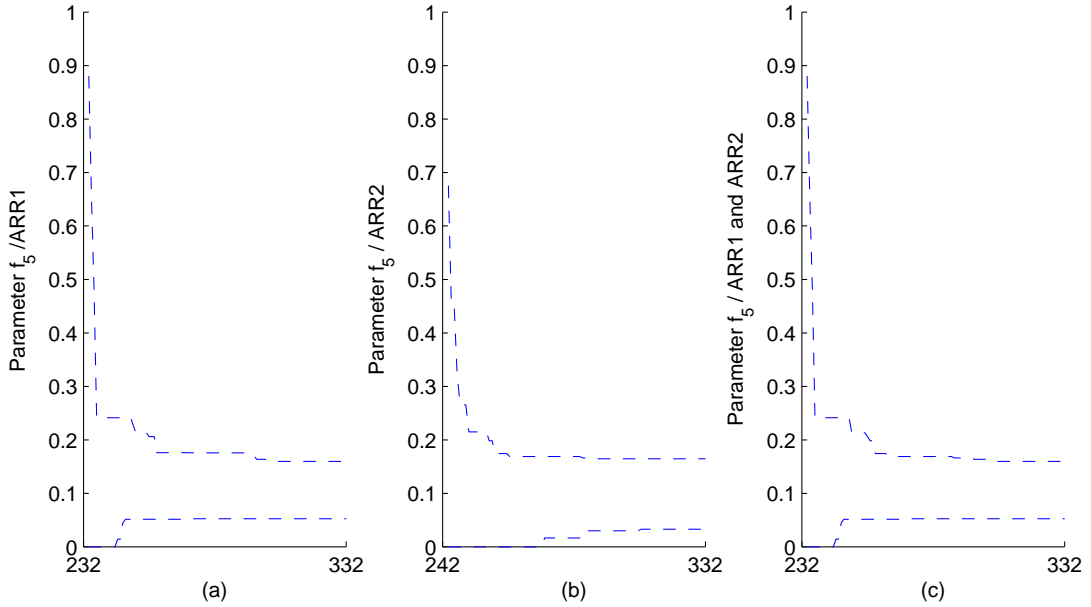


Figure 6.2: Fault identification of parameter  $f_1$ .

Therefore the clogging fault in the output pipe of  $T1$  is the only behavioral mode that can explain the behavior of the system.

Figure 6.3: Fault identification of parameter  $f_5$ .

## 6.4 Conclusions and Contributions

As many engineering problems, fault diagnosis problems can be stated as a CSP in continuous domains. The work presented in this chapter is an example of that. The fault isolation/identification approach presented in this chapter is the third block of the fault diagnosis system that runs on-line: (i) a fault detector which monitors the difference in the observed and expected behavior (residual), (ii) a fault isolation scheme which consists of hypothesis generation and refinement using the fault detection results of the analytical redundancy relations (ARR) of the system, and (iii) the quantitative fault isolation and identification scheme to refine the fault hypothesis test and to estimate the fault magnitude.

The use of overconstrained subsystems of the model, ARRs, leads the approach to tackle the fault identification problem in smaller subproblems which can reduce the complexity of the problem. CSPs are used also to represent the fault identification problem, and hence, to calculate the fault magnitude.

# Chapter 7

## Applications

### Summary

The methods and the theory developed in the previous chapters are applied to several examples. These methods and the corresponding examples are summarized in Table 7.1.

Table 7.1: Application examples.

	SA	FD	QFI	QFII
Positioning control system of an offshore vessel	X			
Electrical distribution systems	X			
Automotive engine	X	X		
DAMADICS Benchmark	X	X	X	
Alcoholic Fermentation Process		X		X

SA: Structural analysis

FD: Fault detection

QFI: Qualitative fault isolation

QFII: Quantitative fault isolation and identification

The analysis of the first and the second applications were developed during my research stay in Denmark, under the supervision of Prof. Mogens Blanke. Moreover, the work done on the automotive engine was developed during a research stay in Sweden, under the supervision of Prof. Erik Frisk.

## 7.1 Positioning Control System of an Offshore Vessel

This application deals with a positioning control system of an offshore vessel. In offshore operations, marine vessels are often required to be kept in a position and heading exclusively by only thrusters known as dynamic positioning (DP) or by mooring system with thruster assistance known as position mooring (PM). A detailed system and model description can be found in [Nguyen & Blanke \(Submitted\)](#); [Nguyen \*et al.\* \(2007\)](#). Dr. Trong Dong Nguyen is acknowledged for providing the model of the position moored tanker.

In this section, the case of a vessel with a four-line mooring system and three thrusters, with notation AUTS, is analyzed. AUTS is a class of the DP systems according the number of thrusters (actuators) and sensors, and measurement units [DNV \(2008\)](#).

The sensors and measurement units include gyros for heading measurements; GPS (Global Positioning System) receivers and hydro-acoustic position reference (HPR) units for position measurements; motion reference units (MRU) for heave, roll, and pitch measurements; anemometers for wind velocity and direction measurements; and a measurement unit for a tension measurement in each mooring line.

There are 27 algebraic and differential constraints describing relationships between a total of 33 variables (Table 7.2). These variables are classified into 3 groups:

- 21 unknown variables,  

$$X = \{T_{T_i}, T_{mo_j}, T_{mo_j}^{xy}, \dot{\nu}, \nu, \ddot{\psi}, \dot{\psi}, \psi, \dot{\mathbf{p}}, \mathbf{p}, \phi, \theta, \mathbf{v}_c, \mathbf{v}_w\}.$$
- 3 known input variables,  

$$K_i = \{u_i\}.$$
- 9 known measured variables,  

$$K_m = \{h, \mathbf{p}_m, T_{mo_m}, \mathbf{hrp}_m, \mathbf{w}_m, \mathbf{c}_m\}.$$

The structure graph corresponding to this model is depicted in Fig. 7.1.

## 7.1 Positioning Control System of an Offshore Vessel

Table 7.2: Structural model for a DP vessel of AUTS class.

Constraint	Component
$a_i : \{u_i, T_{T_i}\}$	thruster, $i = 1 \dots 3$
$c_{2j-1} : \{T_{mo_j}, \mathbf{p}, \psi\}$	mooring line $j = 1 \dots 4$
$c_{2j} : \{T_{mo_j}^{xy}, T_{mo_j}\}$	
$c_9 : \{T_{T_1}, T_{T_2}, T_{T_3}, \mathbf{p}, \psi, \nu, \dot{\nu}, \mathbf{v}_w, \dots, T_{mo_1}^{xy}, \dots, T_{mo_4}^{xy}\}$	surge & sway dynamics
$c_{10} : \{T_{T_1}, T_{T_2}, T_{T_3}, \mathbf{p}, \psi, \ddot{\psi}, \mathbf{v}_w, \dots, T_{mo_1}^{xy}, \dots, T_{mo_4}^{xy}\}$	yaw dynamics
$c_{11} : \{\nu, \dot{\mathbf{p}}, \psi, \mathbf{v}_c\}$	
$d_1 : \{\nu, \dot{\nu}\}$	
$d_2 : \{\mathbf{p}, \dot{\mathbf{p}}\}$	
$d_3 : \{\dot{\psi}, \ddot{\psi}\}$	
$d_4 : \{\psi, \dot{\psi}\}$	
$m_1 : \{\psi, h\}$	gyro
$m_2 : \{\mathbf{p}, \phi, \theta, \psi, \mathbf{p}_m\}$	GPS
$m_3 : \{\phi, \theta, \mathbf{hrp}_m\}$	MRU
$m_4 : \{\mathbf{v}_w, \mathbf{w}_m\}$	anemometer
$m_5 : \{\mathbf{v}_c, \mathbf{c}_m\}$	current velocity measurement
$m_{5+j} : \{T_{mo_j}, T_{mo_{m_j}}\}$	tension measurement $j = 1 \dots 4$

Six MSO sets can be obtained after finding one complete matching over the unknown variables. This set is expanded to a total of 972 MSO sets after applying the algorithm presented in Chapter 3.

Table 7.3 shows the structural detectability and isolability of constraints associated to physical components for a single fault case. As can be seen, faults in  $a_i$  (a thruster), in  $c_{2j}$  (a mooring line), in  $c_{10}$  (yaw dynamics), in  $m_1$  (the gyro), in  $m_4$  (the anemometer), and in  $m_{5+j}$  (a tension measurement unit) are structurally isolable. On the other hand, faults in  $c_9$  (surge & sway dynamics), and in the rest of the sensors and measurement units, i.e.  $m_2$  (GPS),  $m_3$  (MRU), and  $m_5$  (current velocity measurement unit), are only structurally detectable. Faults in  $m_2$  and  $m_3$  are group-wise isolable, i.e. within this group individual faults are detectable but not isolable. Also  $m_5$  and  $c_9$  are group-wise isolable.



## 7.1 Positioning Control System of an Offshore Vessel

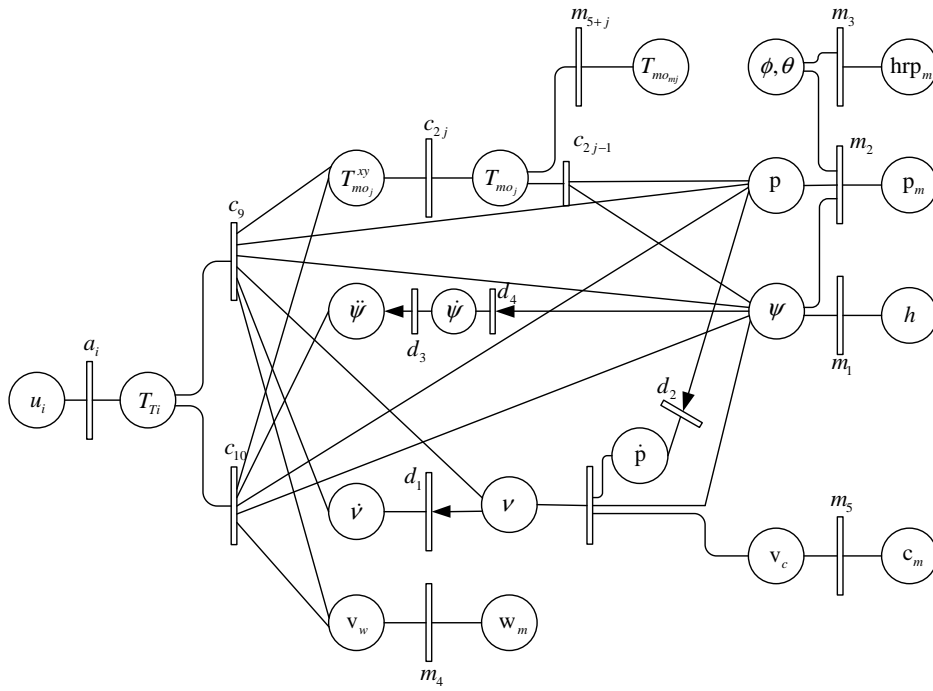


Figure 7.1: Structure graph for the application example.

Table 7.3: Detectability and isolability of single faults.

$a_i$	$c_{2j}$	$c_9$	$c_{10}$	$m_1$	$m_2$	$m_3$	$m_4$	$m_5$	$m_{5+j}$
i	i	d	i	i	d	d	i	d	i

### 7.1.1 Active Structural Isolation

The set of paths through constraints from  $u_i$ , for  $i$  equal to 1 to 3, to the outputs are represented in Table 7.4 of reachability. Following Lemma 3.7.2, it can be deduced that  $\{a_1, a_2, a_3, m_1, m_2, m_6, m_7, m_8, m_9\}$  are structurally isolable when active isolation is employed, while  $c_1, \dots, c_{11}$  remain detectable.

Table 7.4: Output reachability from  $u_i$ .

$u_i \downarrow$	$m_1$	$m_2$	$m_6$	$m_7$	$m_8$	$m_9$	*	**
$h$	1	0	0	0	0	0	1	0
$\mathbf{p}_m$	0	1	0	0	0	0	1	0
$T_{m_1}$	0	0	1	0	0	0	1	0
$T_{m_2}$	0	0	0	1	0	0	1	0
$T_{m_3}$	0	0	0	0	1	0	1	0
$T_{m_4}$	0	0	0	0	0	1	1	0
$\mathbf{w}_m$	0	0	0	0	0	0	0	0
$\mathbf{c}_m$	0	0	0	0	0	0	0	0
$\mathbf{hrp}_m$	0	0	0	0	0	0	0	0

\*:  $\{a_i, c_1, \dots, c_{11}, d_1, \dots, d_4\}$

\*\*: $\{a_j, m_3, m_4, m_5\}, j = \{1, 2, 3\} \wedge j \neq i$

Through this example, algorithm proposed for input to output active structural isolation allows to isolate faults that could only be detectable using a passive approach that includes the set of all MSO sets and the corresponding fault signature matrix. Both the passive and the active techniques can be used in a complementary way to improve the fault detectability and isolability properties. For example, in this application a fault in  $m_2$  (GPS) is structurally isolable when using the active approach but only detectable when using the information provided by all the MSO sets. On the other hand, the opposite occurs when analyzing faults in a mooring line: they are structurally detectable when using the active approach and isolable in the other case.

## 7.2 Applications in Electrical Distribution Systems

In [Knüppel \(2008\)](#), structural analysis for fault detection and isolation in electrical distribution systems, is investigated. In particular, different system representations such as phase components and symmetrical components are applied to examples to analyze the complexity and the completeness of the structural analysis results. A version of SaTool, which is capable of handling loops as well as finding all complete matchings, was used to determine the overall isolability properties for the system. However, because of time and memory limitations, there is a maximum number of matchings that can be found using this version of SaTool. Hence, the isolability properties for some examples were not complete.

In this section, algorithms developed in this thesis using the structural analysis (Chapter 3) are applied to two examples from [Knüppel \(2008\)](#): (i) a nine bus power system, and (ii) a three-phase  $\Delta - Y$  coupled transformer. Mr. Thyge Knüppel is acknowledged for providing the models used in this section.

### 7.2.1 Nine Bus Case Network

The analysis is based on a nine bus power system. It consists of two network components, (1) distribution lines and (2) transformers, which are interconnected at the bus bars in the system. This system has four transformers, six distribution lines and three loads. A single-line diagram of the network is shown in Fig. 7.2.

The feeding station is connected to the distribution grid and serving the local area through two parallel transformers to ensure redundant power supply to the area. Three main stations are serving subjacent feeders where the load is connected, in this example modeled as a load connected directly at the main station. Two local generators are connected to the grid at bus 2 and 3. Further details of the application are presented in [Knüppel \(2008\)](#).

The model has 121 constraints and 140 variables. From these variables, 109 are unknown and 31 known. In the model, the three-phase network was decoupled into three uncoupled sequence networks. With the complete decoupling, the positive-, negative- and zero-sequence system can be analyzed independently.

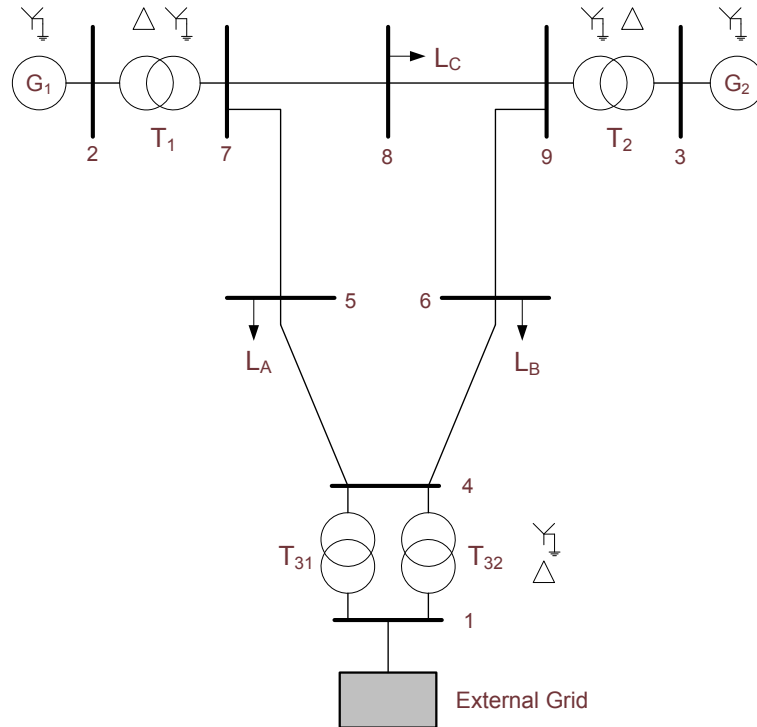


Figure 7.2: Single-line diagram of the nine bus case network.

In this example, the positive- and the negative-sequence network are equivalent and the structural properties for these systems are therefore equivalent. The number of constraints is 38 and the number of unknown variables 35. Using Algorithm 5, proposed in Chapter 3, 78 MSO sets were found in less than a tenth of a second. With regard to the isolability properties, faults in 4 constraints are not detectable (as obtained in Knüppel (2008)), faults in 9 constraints are isolable, and faults in 25 constraints are group-wise isolable.

The model of the zero-sequence has 45 constraints and 39 unknown variables. In this case, 2579 MSO sets were found in seconds. Regarding the isolability properties, faults in 8 constraints are not detectable (as obtained in Knüppel (2008)), faults in 13 constraints are isolable, and faults in 24 constraints are group-wise isolable (within groups of maximum 3 constraints, faults are detectable but not isolable).

Comparing the results presented in Knüppel (2008), the isolability properties

were improved in this work since all MSO sets were found.

### 7.2.2 Three-phase $\Delta - Y$ Coupled Transformer

This system is composed of a three-phase  $\Delta - Y$  coupled transformer. The transformer has its receiving end connected to a load modeled as three known single-phase ohmic loads. The currents at the sending end are controlled and the voltages at the sending end are measured.

Various representations are analyzed in Knüppel (2008). In this thesis, the composite notation (i.e. a notation that combines scalar and vector notations) is used to represent the model of the system. This system has 19 variables with 2 known, and 18 constraints from which 4 cannot fail (the constraints that represent the relation between a variable and its derivative). Moreover, causality information is added to the structural model.

Table 7.5: Incidence matrix for the system represented with phase components in composite notation.

	$y_1$	$y_2$	$y_3$	$y_4$	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$	$x_8$	$x_9$	$x_{10}$	$x_{11}$	$x_{12}$	$x_{13}$	$x_{14}$	$x_{15}$	
$c_1$	0	0	0	0	1	x	0	0	0	0	0	0	0	0	0	0	0	0	0	
$c_2$	0	0	0	0	0	0	1	x	1	0	0	0	0	0	0	0	0	0	0	
$c_3$	0	0	0	0	0	0	1	0	0	1	1	1	0	0	0	0	0	0	0	
$c_4$	0	0	0	0	0	0	0	1	0	0	0	0	1	1	1	0	0	0	0	
$c_5$	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	x	1	0	
$c_6$	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	1	
$c_7$	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	x
$c_8$	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
$c_9$	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$c_{10}$	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$c_{11}$	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
$c_{12}$	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
$c_{13}$	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0
$c_{14}$	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0
$c_{15}$	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	x	0
$c_{16}$	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	x
$c_{17}$	0	0	0	0	0	0	x	0	0	0	0	1	0	0	0	0	0	0	0	0
$c_{18}$	0	0	0	0	0	0	0	x	0	0	0	0	0	0	1	0	0	0	0	0

## 7.2 Applications in Electrical Distribution Systems

---

Using the algorithm proposed in Chapter 3, 17 MSO sets were found. Of these 17 MSO sets, 15 have at least one causal model. Results including all the causal models are obtained in a few tenths of a second. With regard to the isolability properties, faults in 6 constraints are isolable, and faults in 7 constraints are group-wise isolable. Same results were obtained in Knüppel (2008) using SaTool, but after finding all complete matchings (more than 3500).

Table 7.6: Detectability and isolability in a single fault case.

$c_1$	$c_2$	$c_3$	$c_4$	$c_5$	$c_6$	$c_7$	$c_8$	$c_9$	$c_{10}$	$c_{11}$	$c_{12}$	$c_{13}$	$c_{14}$
$i$	0	$d_\diamond$	$d_\square$	$d_\diamond$	$d_\square$	$d_\diamond$	$i$	$i$	$i$	$i$	$i$	$d_\diamond$	$d_\square$

$d_\square$  and  $d_\diamond$  indicate faults that are detectable but group-wise isolable.

## 7.3 An Automotive Engine

Automotive engines is an important application for model-based diagnosis not only because of environmentally based legislative regulations, but also because of reparability, availability, and vehicle protection Nyberg (2002). Different model-based approaches have been studied in several works as in Gertler *et al.* (1995); Kimmich *et al.* (2005); Nyberg (2002); Nyberg & Nielsen (1998); Nyberg *et al.* (2001). Possible faults include sensor faults, actuator faults and leakages, e.g. boost leakage, manifold leakage, pressure sensor bias, pressure sensor gain-fault. These types of faults typically lead to degraded emission control, and also possible damage to engine components.

### 7.3.1 Structural Analysis

The model used in this section is a part of the Mean Value Engine Model explained in Andersson (2005). This model describes the average behavior of the engine over one to several thousands of engine cycles, and is a component based model in which each component is described in terms of equations, constants, parameters, states, inputs and outputs. A schematic picture of the system system is shown in Fig. 7.3.

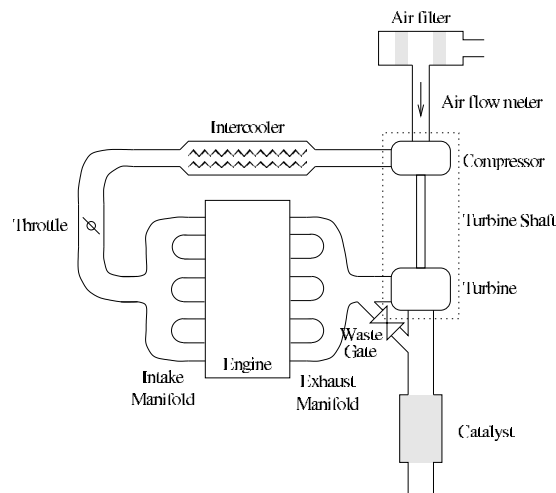


Figure 7.3: A schematic figure of the turbo-charged engine.

### 7.3 An Automotive Engine

The main components are: air-filter, compressor, intercooler, throttle, engine, turbine, and exhaust system. Table 7.7 shows a list of variables that are measurable.

Table 7.7: Measurable variables.

<i>Pressure sensors</i>		<i>Temperature sensors</i>	
<i>Location</i>	<i>Symbol</i>	<i>Location</i>	<i>Symbol</i>
Ambient pressure	$p_a$	Air-Filter entry	$T_a$
Before compressor	$p_{af}$	After compressor	$T_{comp}$
After compressor	$p_{comp}$	After intercooler	$T_{ic}$
After intercooler	$p_{ic}$	In intake manifold	$T_{im}$
Intake manifold	$p_{im}$	Exhaust manifold	$T_{em}$
Exhaust manifold	$p_{em}$	After turbine	$T_t$
After Turbine	$p_t$		

<i>Miscellaneous sensors</i>			
<i>Location</i>	<i>Symbol</i>	<i>Location</i>	<i>Symbol</i>
Air-mass flow	$W_{af}$	Air-fuel ratio	$\lambda$
Throttle angle	$\alpha$	Torque	$T_q$
Engine speed	$N$	Injection time	$t_{inj}$
Turbine speed	$N_t$		

During my research stay in Sweden, a simulation model of the system was implemented in EcosimPro [EA Internacional \(2008\)](#). EcosimPro is a simulation environment for complex hybrid systems, represented by differential-algebraic equations (DAE) or ordinary-differential equations (ODE) and discrete events.

Due to the complexity of the engine model, a routine was programmed in Matlab to automatically transfer the EcosimPro model to a structural model. Figure 7.4 shows the incidence matrix for the system. This model has 104 constraints and 100 unknown variables. From the set of measurable variables presented in Table 7.7, it is assumed that a subset of 10 sensors are part of the diagnosis system. These sensors measure: throttle angle  $\alpha$ , ambient pressure  $p_a$ , ambient temperature  $T_a$ , pressure  $p_{im}$  and temperature  $T_{im}$  in the intake manifold, pressure  $p_{em}$  and temperature  $T_{em}$  in the exhaust manifold, pressure  $p_{ic}$  and temperature  $T_{ic}$  after the intercooler, and engine speed  $N$ .



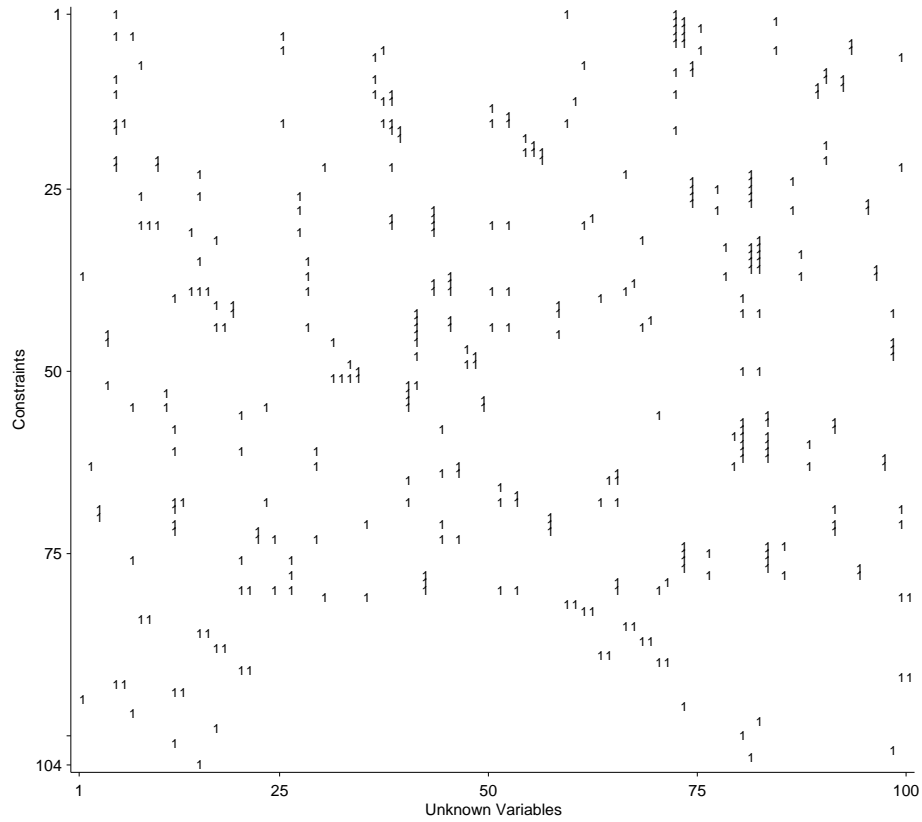


Figure 7.4: Incidence matrix of the engine structural model.

Algorithm 5, described in Chapter 3, is used to demonstrate the efficiency of it when applying to a real-world example. After finding one complete matching, 4 basic MSO sets are found. The combination of these sets leads to 2949 MSO sets (see Table 7.8).

Table 7.8: Results of Algorithm 5 applied to the application example

Collection	Number of MSO sets
$\mathcal{C}_{MSO_1}$	4
$\mathcal{C}_{MSO_2}$	76
$\mathcal{C}_{MSO_3}$	702
$\mathcal{C}_{MSO_4}$	2167
$\mathcal{C}_{MSO}$	2949

### 7.3.2 Fault Detection in the Air Intake System

One important part of the diagnosis requirements for automotive engines is the air path. In this section of the example, the problem of fault detection in the air intake system is stated as a constraint satisfaction problem and solved using interval-based consistency techniques (see Chapter 4).

In the air intake system, ambient air enters the system and an air-mass flow sensor measures the air-mass flow rate  $W_a$ . Next, the air passes the compressor side of the turbo-charger, the intercooler and then the throttle. The flow  $W_{th}$  is dependent on the intercooler and manifold pressures,  $p_{ic}$  and  $p_{im}$ , the temperature  $T_{ic}$ , and the throttle angle  $\alpha$ . Finally the air enters the cylinder and this flow,  $W_{cyl}$  is dependent on  $p_{im}$  and  $p_{em}$ , the temperature  $T_{im}$ , the engine speed  $N$  and the air-fuel ratio  $\lambda$ .

The equations describing the fault free air intake model can be written as

$$\frac{dp_{im}}{dt} = \frac{R_a T_{im}}{V_{im}} (W_{th} - W_{cyl}) + \frac{m_{im} R_a}{V_{im}} \frac{dT_{im}}{dt} \quad (7.1)$$

$$W_{th} = \frac{p_{ic}}{\sqrt{R_a T_{ic}}} \Psi(\Pi) A_{eff}(\alpha) \quad (7.2)$$

$$W_{cyl} = p_{im} C_1 \frac{1}{1 + \frac{1}{\lambda(\frac{A}{P})_s}} \frac{r_c - (\frac{p_{em}}{p_{im}})^{\frac{1}{\gamma_a}}}{r_c - 1} V_d \frac{N}{R_{im} T_{im}} \quad (7.3)$$

where

$$\Pi = \frac{p_{im}}{p_{ic}} \quad (7.4)$$

$$\Psi^*(\Pi) = \sqrt{\frac{2\gamma}{\gamma-1} (\Pi^{\frac{2}{\gamma}} - \Pi^{\frac{\gamma+1}{\gamma}})} \quad (7.5)$$

$$\Psi(\Pi) = \begin{cases} \sqrt{\gamma \left(\frac{2}{\gamma+1}\right)^{\frac{\gamma+1}{\gamma-1}}} & 0 < \Pi \leq \left(\frac{2}{\gamma+1}\right)^{\frac{\gamma}{\gamma+1}} \\ \Psi^*(\Pi) & \left(\frac{2}{\gamma+1}\right)^{\frac{\gamma}{\gamma+1}} < \Pi \leq \Pi_{lin} \\ \frac{\Psi^*(\Pi_{lin})}{\Pi_{lin}-1} (\Pi - 1) & \Pi_{lin} < \Pi \leq 1 \end{cases} \quad (7.6)$$

The interval method presented in this paper uses discrete-time models, in this case a discretization is obtained by using a first order approximation:

$$\mathbf{x}(t + T_s) \simeq \mathbf{x}(t) + T_s \mathbf{g}(\mathbf{x}(t), \mathbf{u}(t), \boldsymbol{\theta}), \quad (7.7)$$

where the sample time,  $T_s$ , is equal to 10ms.

Thus, from Eq. 7.1 and including a non-linear interval observer (see Section 4.3.1.1), it is obtained:

$$\begin{aligned} \hat{p}_{im}(k+1) &= \hat{p}_{im}(k) + T_s \frac{R_a T_{im}(k)}{V_{im}} (\hat{W}_{th}(k) - \hat{W}_{cyl}(k)) \\ &\quad + K(p_{im}(k) - \hat{p}_{im}(k)) \end{aligned} \quad (7.8)$$

where the set of sensors considered are: pressures  $p_{im}$ ,  $p_{ic}$  and  $p_{em}$ , temperatures  $T_{im}$  and  $T_{ic}$ , engine speed  $N$  and throttle plate angle  $\alpha$ .

The uncertain parameters selected are two engine specific parameters, and those are the gain parameter  $C_1$ , which describes the engine pumping capabilities, and the ratio of specific heats  $\gamma$ . They have been bounded using the criterion that in the fault free case, there should be no false alarm. The variable  $\lambda$  (the air-fuel ratio) has been considered as an interval, instead of the measured value, because of the accuracy of the sensor and for a sake of simplicity.

The set of variables of this model represented as a CSP is

$$\begin{aligned} \mathcal{V} &= \{C_1, \gamma, \lambda(k-w), \dots, \lambda(k-1), \hat{p}_{im}(k-w), \dots, \hat{p}_{im}(k), \\ &\quad p_{im}(k-w), \dots, p_{im}(k-1), p_{ic}(k-w), \dots, p_{ic}(k-1), \\ &\quad p_{em}(k-w), \dots, p_{em}(k-1), T_{ic}(k-w), \dots, T_{ic}(k-1), \\ &\quad T_{im}(k-w), \dots, T_{im}(k-1), N(k-w), \dots, N(k-1), \\ &\quad \alpha(k-w), \dots, \alpha(k-1)\}, \end{aligned}$$

and the set of initial domains for the estimated variable  $\hat{p}_{im}$  has been taken equal to  $[1 * 10^4, 2 * 10^5]$  with the exception of the initial domains of  $\hat{p}_{im}(k-w)$  and  $\hat{p}_{im}(k)$ , at the beginning and the end of the time window, which have been assigned a value equal to the interval measurements  $p_{im}(k-w)$  and  $p_{im}(k)$ .

### 7.3.3 Experimental Results: Detection

All experiments were performed on a four-cylinder turbo-charged spark-ignited SAAB engine located in the research laboratory at Vehicular Systems Group, Linköping University. The engine is mounted in a test bench together with a Schenck dynamometer.

In this section two faulty scenarios are considered, (i) a gain-fault in the sensor of pressure  $p_{ic}$ , and (ii), a gain-fault in the sensor of pressure  $p_{im}$ . The fault detection results are obtained by using Weak-3B consistency technique [Granvilliers \(2004\)](#) and a window length equal to 30 samples (0.3 s). The required computation time and the sample time have the same order of magnitude, thus, allowing real-time implementations.

When no solution is found to the CSP, a fault is detected. Otherwise, when the observed behavior and the model are not proven to be inconsistent, means either that there is not a fault or that it could not be detected. In this way, the proposed approach prioritizes to avoid false alarms to missed alarms.

#### 7.3.3.1 First Scenario

In Fig. 7.5, obtained results in the case of no fault and a 10% gain-fault in the pressure sensor of  $p_{ic}$  are shown. A “1” indicates there is a fault and a “0” means there is not a fault or it could not be detected. As shown in this figure, there is no false alarm in absence of fault. The fault in the sensor begins at sample 600 and is detected at sample 604.

Figure 7.6 shows the interval measurement (solid line) and the estimated manifold pressure (dashed line) in the fault free situation. The external estimate has been obtained with the domains for the estimated variable  $\hat{p}_{im}(k)$  equal to  $[1 \cdot 10^4, 2 \cdot 10^5]$  in the CSP representation presented in Section 7.3.2. Although the computation time is bigger than the sample time being not suitable to operate in real-time, it can be used when a fault is detected to obtain more information, and then, to improve the task of diagnosis (Section 7.3.4).

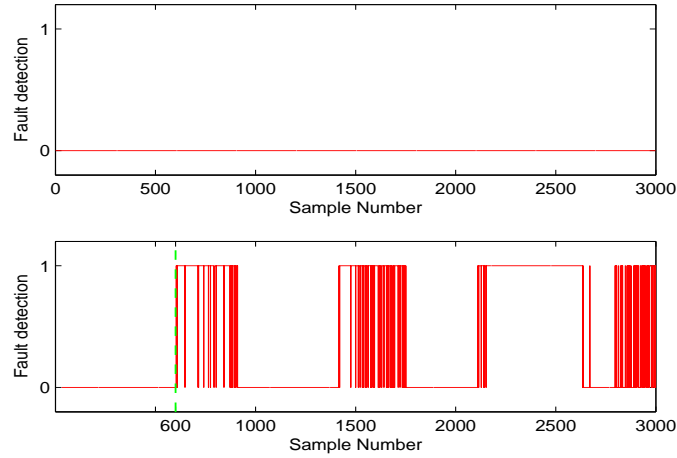


Figure 7.5: First scenario fault detection. Top: no fault. Bottom: gain-fault in the sensor of pressure  $p_{ic}$  beginning at sample 600. The fault is detected from sample 604.

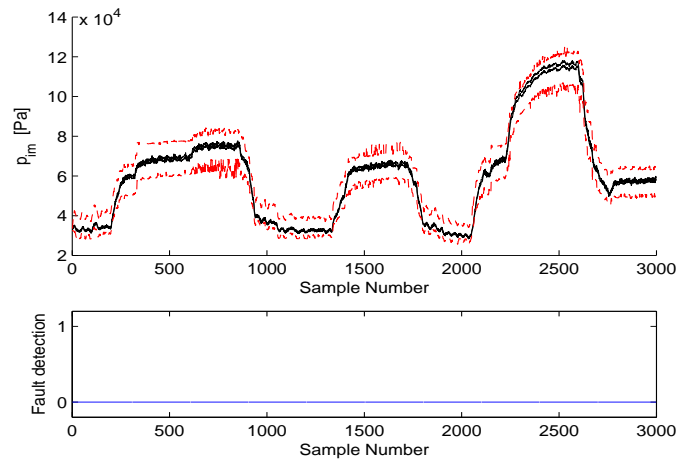


Figure 7.6: First scenario without faults. The upper plot shows measured and estimated manifold pressure.

### 7.3.3.2 Second Scenario

Figure 7.7 shows the results in the case of no fault and a 10% gain-fault in the pressure sensor of  $p_{im}$ . The fault in the sensor begins at sample 800 and is detected at the same time as the fault appears.

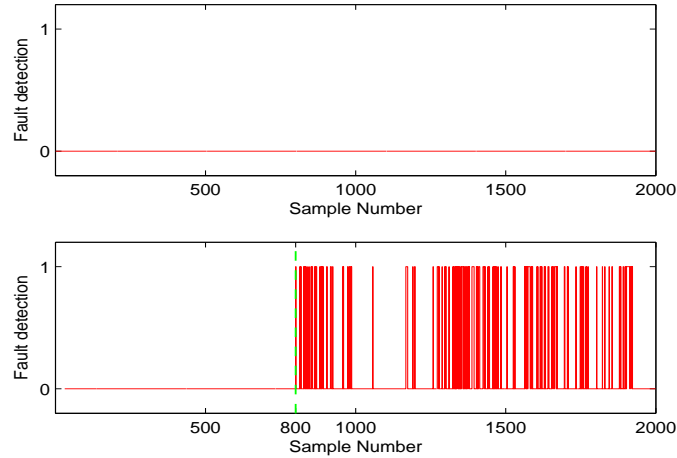


Figure 7.7: Second scenario fault detection. Top: no fault. Bottom: gain-fault in the sensor of pressure  $p_{im}$ . The fault is detected at the same time as the fault appears.

Figure 7.8 shows the interval measurement and the estimated manifold pressure in the fault free situation of this scenario.

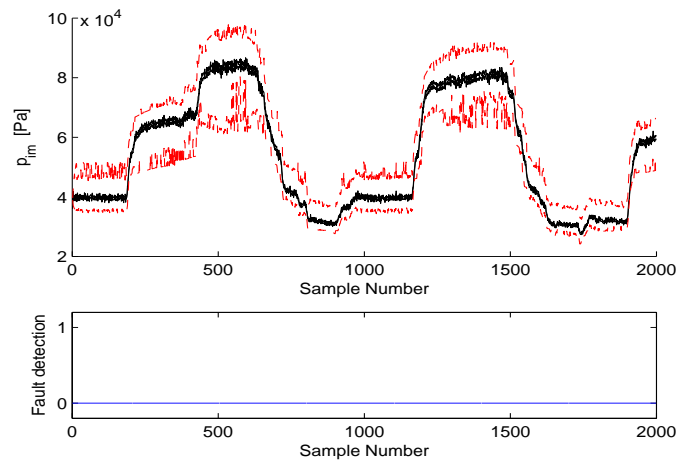


Figure 7.8: Second scenario without faults. The upper plot shows measured and estimated manifold pressure.

### 7.3.4 Diagnosis: Signs of the Symptoms

When it is possible to utilize detailed models for the faults, this information can be used together with the signs in the residuals, to prune the candidate space when performing the fault diagnosis task, as proposed in Chapter 5.

This approach could be applied to perform the diagnosis in both studied scenarios. In order to do this, it is needed to:

- Include in the fault signature matrix, the influence of the faults in the residuals, and
- Obtain the sign of the symptom. This could be obtained by observing the behavior of the estimated output with respect to the measurement. For instance, the sign would be +1 if the estimation is greater than the interval measurement, or if the estimation is smaller than the interval measurement, the sign would be -1.

For both scenarios, when a fault is detected, the algorithm estimates the manifold pressure at the end of each sliding window and the consistent region of this variable can be seen in Figs. 7.9 and 7.10. As it is expected, the interval measurement (solid line) does not intersect with the estimate (dashed line), and for the first case, the estimates are always smaller than the measurements, whereas for the second case, the opposite relation is observed. This is the information that should be compared with the theoretical signatures with the signs of symptoms, as explained in Chapter 5.

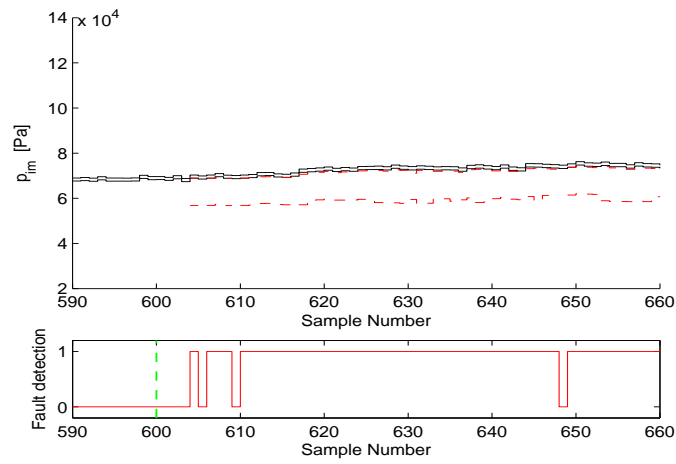


Figure 7.9: First scenario with a gain-fault in the sensor of pressure  $p_{ic}$ . Starts at 600, detected at 604.

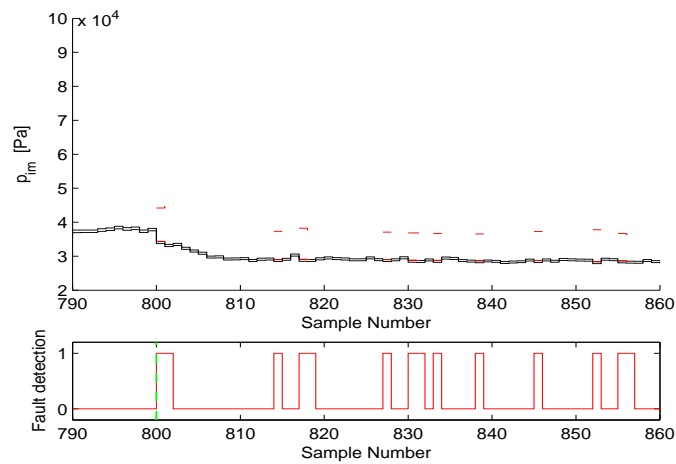


Figure 7.10: Second scenario with a gain-fault in the sensor of pressure  $p_{im}$ . Starts at 800, detected at the same time as the fault.



## 7.4 DAMADICS Benchmark

This application example deals with an industrial smart actuator, proposed as an FDI benchmark in the European project DAMADICS. As described in [Bartyś \*et al.\* \(2006\)](#), the valve consists of three main components:

- Control and by-pass valves. The control valve acts on the flow of the fluid passing through the pipeline installation.
- Spring-and-diaphragm pneumatic servo-motor. This is a compressible fluid powered device in which the fluid acts upon the flexible diaphragm, to provide linear motion of the servomotor stem.
- Positioner. The positioner is a device applied to eliminate the control-valve-stem miss-positions produced by external or internal sources such as: friction, clearance in mechanical assemblies, supply pressure variations, hydrodynamic forces, etc.

A general scheme of the valve is shown in Fig. 7.11.

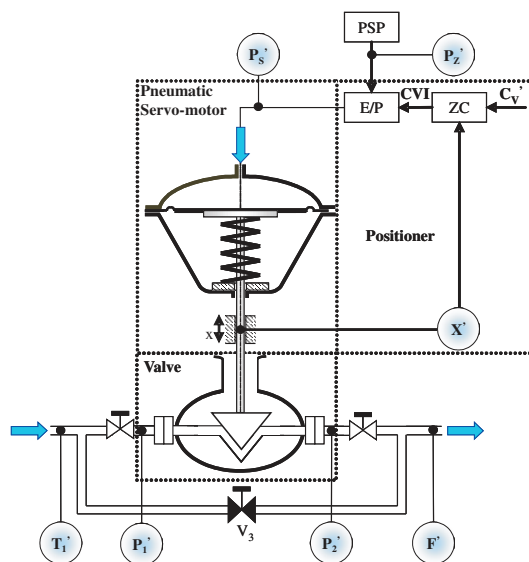


Figure 7.11: General scheme of the DAMADICS valve.

Measured variables, indicated by circles in the figure, are the valve plug position  $X'$ , the fluid flow  $F'$ , fluid temperature  $T'_1$ , upstream and downstream pressure of the valve  $P'_1$ ,  $P'_2$ , the transducer chamber pressure  $P'_s$  and the positioner supply pressure  $P'_z$ .

The actuator model is based on physical description and engineering expertise, and is implemented in the Matlab-Simulink environment [Bartyś & de las Heras \(2003\)](#). It can be used to simulate up to 19 incipient or abrupt faults. The set of faults acting on the valve and its components is described in [Bartyś & Syfert \(2002\)](#). In this thesis, the set of faults was extended by 4 additional faults to consider all measurement faults. The fault range for each fault  $f_i$  is standardized to the ranges  $< 0, 1 >$ , or  $< -1, 1 >$  when the fault may cause bi-directional consequences, e.g. sensor faults.

### 7.4.1 Residual Generation

The number of equations, transferred from the Simulink Model, is 15, and the number of unknown variables is 11. The incidence matrix with unknown variables is presented in Table 7.9. The edges of a matching are identified by a “Ⓛ” in the matrix.

Table 7.9: Incidence Matrix of the valve.

	$X$	$P_s$	$F_{vc}$	$P_1$	$P_2$	$T_1$	$F_v$	$P_z$	$CVI$	$F$	$F_{v3}$
$c_1$	1	1	1								
$c_2$	1		Ⓛ	1	1	1					
$c_3$	1			1	1	1	Ⓛ				
$c_4$	1	1						1	1		
$c_5$									1		
$c_6$							1			1	1
$c_7$				1	1	1					Ⓛ
$c_8$	Ⓛ										
$c_9$										Ⓛ	
$c_{10}$						Ⓛ					
$c_{11}$				Ⓛ							
$c_{12}$					Ⓛ						
$c_{13}$								Ⓛ			
$c_{14}$		Ⓛ									
$c_{15}$									Ⓛ		

The algorithm presented in Chapter 3 leads to 29 MSO sets, summarized in Table 7.10, after combining 4 basic MSO sets (see Table 7.11).

Table 7.10: MSO sets of the application example.

$n^o$	MSO sets
1	$\{c_1, c_2, c_8, c_{10}, c_{11}, c_{12}, c_{14}\}$
2	$\{c_4, c_8, c_{13}, c_{14}, c_{15}\}$
3	$\{c_5, c_{15}\}$
4	$\{c_3, c_6, c_7, c_8, c_9, c_{10}, c_{11}, c_{12}\}$
5	$\{c_1, c_2, c_4, c_{10}, c_{11}, c_{12}, c_{13}, c_{14}, c_{15}\}$
6	$\{c_1, c_2, c_4, c_8, c_{10}, c_{11}, c_{12}, c_{13}, c_{15}\}$
7	$\{c_1, c_2, c_3, c_6, c_7, c_9, c_{10}, c_{11}, c_{12}, c_{14}\}$
8	$\{c_1, c_2, c_3, c_6, c_7, c_8, c_9, c_{11}, c_{12}, c_{14}\}$
9	$\{c_1, c_2, c_3, c_6, c_7, c_8, c_9, c_{10}, c_{12}, c_{14}\}$
10	$\{c_1, c_2, c_3, c_6, c_7, c_8, c_9, c_{10}, c_{11}, c_{14}\}$
11	$\{c_4, c_5, c_8, c_{13}, c_{14}\}$
12	$\{c_1, c_2, c_4, c_5, c_{10}, c_{11}, c_{12}, c_{13}, c_{14}\}$
13	$\{c_1, c_2, c_4, c_5, c_8, c_{10}, c_{11}, c_{12}, c_{13}\}$
14	$\{c_3, c_4, c_5, c_6, c_7, c_9, c_{10}, c_{11}, c_{12}, c_{13}, c_{14}\}$
15	$\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9, c_{10}, c_{11}, c_{13}\}$
16	$\{c_3, c_4, c_6, c_7, c_9, c_{10}, c_{11}, c_{12}, c_{13}, c_{14}, c_{15}\}$
17	$\{c_1, c_2, c_3, c_4, c_6, c_7, c_8, c_9, c_{11}, c_{12}, c_{13}, c_{15}\}$
18	$\{c_1, c_2, c_3, c_4, c_6, c_7, c_8, c_9, c_{10}, c_{12}, c_{13}, c_{15}\}$
19	$\{c_1, c_2, c_3, c_4, c_6, c_7, c_8, c_9, c_{10}, c_{11}, c_{13}, c_{15}\}$
20	$\{c_1, c_2, c_3, c_4, c_6, c_7, c_9, c_{11}, c_{12}, c_{13}, c_{14}, c_{15}\}$
21	$\{c_1, c_2, c_3, c_4, c_6, c_7, c_9, c_{10}, c_{12}, c_{13}, c_{14}, c_{15}\}$
22	$\{c_1, c_2, c_3, c_4, c_6, c_7, c_9, c_{10}, c_{11}, c_{13}, c_{14}, c_{15}\}$
23	$\{c_1, c_2, c_3, c_4, c_6, c_7, c_9, c_{10}, c_{11}, c_{12}, c_{13}, c_{15}\}$
24	$\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9, c_{11}, c_{12}, c_{13}\}$
25	$\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9, c_{10}, c_{12}, c_{13}\}$
26	$\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_9, c_{11}, c_{12}, c_{13}, c_{14}\}$
27	$\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_9, c_{10}, c_{12}, c_{13}, c_{14}\}$
28	$\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_9, c_{10}, c_{11}, c_{13}, c_{14}\}$
29	$\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_9, c_{10}, c_{11}, c_{12}, c_{13}\}$

Table 7.11: Results of Algorithm 2 applied to the application example

Collection	Number of MSO sets
$\mathcal{C}_{MSO_1}$	4
$\mathcal{C}_{MSO_2}$	8
$\mathcal{C}_{MSO_3}$	10
$\mathcal{C}_{MSO_4}$	7
$\mathcal{C}_{MSO}$	29

The fault isolability analysis matrix for the 4 basic MSO sets is given as shown in Table 7.12. From this table, it is seen that the first five blocks of 4, 5, 5, 3 and 2 faults, respectively, show faults that are group-wise isolable, i.e. they are not isolable from the other faults in the group, but isolable from other faults in different groups.  $f_{13}$ ,  $f_{15}$  and  $f_{23}$  are isolable from all other faults.

Table 7.12: Fault isolability analysis matrix.

	$f_{11}$	$f_8$	$f_4$	$f_1$	$f_{22}$	$f_{21}$	$f_{20}$	$f_3$	$f_2$	$f_{19}$	$f_{18}$	$f_7$	$f_6$	$f_5$	$f_{16}$	$f_{12}$	$f_9$	$f_{14}$	$f_{10}$	$f_{13}$	$f_{15}$	$f_{23}$
$f_{11}$	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$f_8$	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$f_4$	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$f_1$	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$f_{22}$	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
$f_{21}$	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
$f_{20}$	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
$f_3$	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
$f_2$	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
$f_{19}$	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0
$f_{18}$	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0
$f_7$	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0
$f_6$	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0
$f_5$	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0
$f_{16}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0
$f_{12}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0
$f_9$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0
$f_{14}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0
$f_{10}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0
$f_{13}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
$f_{15}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
$f_{23}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

When considering the complete collection of MSO sets, a fault isolability matrix as in Table 7.13 is obtained. As it can be seen, now it is possible to isolate  $f_{10}$ ,  $f_{14}$ ,  $f_{20}$ ,  $f_{21}$  and  $f_{22}$  from each other. Moreover,  $f_2$  is not isolable from  $f_3$  and  $f_3$  is not isolable from  $f_2$ .

Table 7.13: Fault isolability analysis matrix using the 29 MSO sets.

	$f_{11}$	$f_8$	$f_4$	$f_1$	$f_{22}$	$f_{21}$	$f_{20}$	$f_3$	$f_2$	$f_{19}$	$f_{18}$	$f_7$	$f_6$	$f_5$	$f_{16}$	$f_{12}$	$f_9$	$f_{14}$	$f_{10}$	$f_{13}$	$f_{15}$	$f_{23}$
$f_{11}$	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$f_8$	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$f_4$	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$f_1$	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$f_{22}$	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$f_{21}$	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$f_{20}$	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$f_3$	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
$f_2$	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
$f_{19}$	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0
$f_{18}$	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0
$f_7$	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0
$f_6$	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0
$f_5$	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0
$f_{16}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0
$f_{12}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0
$f_9$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0
$f_{14}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
$f_{10}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
$f_{13}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
$f_{15}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
$f_{23}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

### 7.4.2 Fault Detection

In this section, two ARRs are used to illustrate the proposed approach: a static ARR (derived from the fourth MSO set of Table 7.10),  $r_1$ , and a dynamic ARR,  $r_2$  (derived from the first MSO set). The first residual, Eq. 7.9, can be generated from fluid flow equation, and the second residual, Eq. 7.13, from rod displacement equation (note that  $\hat{X}$  is calculated using a dynamic force balance equation).

The computational form of the first residual is given by:

$$r_1^{comp} = F' - \hat{F}_v - \hat{F}_{v3} \tag{7.9}$$

where  $F_v$  is the flow through the control valve,  $\Delta p$  is the difference between up- and downstream valve pressures, and  $F_{v3}$  is the flow through the bypass valve V3.

$$\hat{F}_v = \rho(T'_1, P'_1) * K_v(X') * 100 * \sqrt{\frac{\Delta p}{\rho(T'_1, P'_1)}} \quad (7.10)$$

$$\Delta p = \min(P'_1 - P'_2, K_m(X') * (P'_1 - r_c(P'_1) * p_v(T'_1))) \quad (7.11)$$

$$\hat{F}_{v3} = \rho(T'_1, P'_1) * K_{v3}(X'_3) * \sqrt{\frac{P'_1 - P'_2}{\rho(T'_1, P'_1)}} \quad (7.12)$$

The computational form of the second residual is given by:

$$r_2^{comp} = X' - \hat{X}(P'_s, \hat{F}_{vc}) \quad (7.13)$$

where the opposing force  $F_{vc}$  (the vena-contracta force) is:

$$\hat{F}_{vc} = \pi * r^2 * \max\left(\frac{P'_1 - P'_2}{K_m(X')}, p_v(T'_1)\right) \quad (7.14)$$

In this application, only single faults are taken into account,  $f_i f_j = 0, \forall i \neq j$ . Considering residuals generated by Eqs. 7.9 and 7.13, the theoretical fault signature matrix presented in Table 7.14 can be deduced. A ‘1’ appears when the corresponding fault affects at least one elementary relation used to generate the residual. It means that the considered fault ideally influences the ARR.

Table 7.14: Fault signature matrix

	$f_1$	$f_2$	$f_3$	$f_4$	$f_5$	$f_6$	$f_7$	$f_8$	$f_9$	$f_{10}$	$f_{11}$	$f_{12}$	$f_{13}$	$f_{14}$	$f_{15}$	$f_{16}$	$f_{17}$	$f_{18}$	$f_{19}$
$r_1$	0	1	1	0	1	1	1	0	0	0	0	0	1	0	0	0	0	1	1
$r_2$	1	1	1	1	0	0	0	1	0	1	1	0	1	1	0	0	0	0	0

For instance, the sensitivity of the first ARR,  $r_1$ , with respect to the fault  $f_5$  is presented in Eq. 7.17, and it is obtained by using the model of the ARR that includes the fault (Eq. 7.15), and in particular, the corresponding internal form (Eq. 7.16). Since  $k_{f_5} > 0$  and  $\hat{F}_v > 0$ , the sign of  $s_{15}$  is  $-1$ .

$$r_1 = F' - \hat{F}_v * (1 - k_{f5} * f_5) - \hat{F}_{v3} \quad (7.15)$$

$$r_1^{int} = -\hat{F}_v * k_{f5} * f_5 \quad (7.16)$$

$$s_{15} = \frac{\partial r_1^{int}}{\partial f_5} = -\hat{F}_v * k_{f5} \quad (7.17)$$

The sign of the ARR by considering the possible variation in the sign of each fault is analyzed for each case in Table 7.15. For instance, for  $r_1$  and  $f_{19}$ , if  $f_{19}$  is positive, then the symptom of  $r_1$  will be negative. Consequently, if  $f_{19}$  is negative, then the symptom of  $r_1$  will be positive. This behavior is reflected in the table using  $\mp 1$ . Symbol ‘1’ denotes that the sign could be positive or negative, for instance  $f_4$  and  $f_8$  modify the hysteresis loop and then real value of  $X$  will be greater than, or less than the estimated value depending on the movement direction of  $X$ , and the sign of the sensitivity cannot be completed beforehand.

Table 7.15: ARRs and their related fault modes using the sign of the symptom

	$f_1$	$f_2$	$f_3$	$f_4$	$f_5$	$f_6$	$f_7$	$f_8$	$f_9$	$f_{10}$	$f_{11}$	$f_{12}$	$f_{13}$	$f_{14}$	$f_{15}$	$f_{16}$	$f_{17}$	$f_{18}$	$f_{19}$
$r_1$	0	-1	+1	0	-1	+1	-1	1	0	0	0	0	$\pm 1$	0	0	0	0	+1	$\mp 1$
$r_2$	-1	-1	+1	1	0	0	0	1	0	-1	+1	0	$\pm 1$	$\pm 1$	0	0	0	0	0

### 7.4.3 Diagnosis Results

Table 7.16 shows the sets of possible diagnostics obtained using the DX approach (row analysis of the fault signature matrix) when the sign of the residuals is used, and when it is not used. The signs help with discarding some diagnostics, and so the sets are reduced. In this example only two ARRs have considered, but the reduction is larger if more ARRs are used.

### 7.4.4 Simulation Results

A faulty scenario involving an incipient fault,  $f_{13}$ , in the rod displacement sensor is considered. The values of the variables are represented by intervals to take into account any associated uncertainty in the measurements. The parameters of the

Table 7.16: Diagnosis using and not using signs

Symptoms		Diagnosis	Diagnosis using signs
$\text{sgn}(r_1)$	$\text{sgn}(r_2)$		
-1	-1	$f_2, f_3, f_8, f_{13}$	$f_2, f_8, f_{13}$
-1	+1		$f_8$
+1	-1		$f_8$
+1	+1		$f_3, f_8, f_{13}$
-1	0	$f_2, f_3, f_5, f_6, f_7, f_8, f_{13}, f_{18}, f_{19}$	$f_2, f_5, f_7, f_8, f_{13}, f_{19}$
+1	0		$f_3, f_6, f_8, f_{13}, f_{18}, f_{19}$
0	-1	$f_1, f_2, f_3, f_4, f_8, f_{10}, f_{11}, f_{13}, f_{14}$	$f_1, f_2, f_4, f_8, f_{10}, f_{13}, f_{14}$
0	+1		$f_3, f_4, f_8, f_{11}, f_{13}, f_{14}$

model are also taken as intervals for the same reason. Some parameters like the density  $\rho$ , which is a function of the measurements  $T'_1$  and  $P'_1$ , are considered by using intervals to reduce the complexity to compute the residual.

A window from SQualTrack for the models of  $r_1$  and  $r_2$  is shown in Fig. 7.12 and 7.13 when  $f_{13}$  occurs. The upper graphs show the envelopes of the output variable (the inner envelope in dotted lines, and the outer envelope in dashed lines), and the interval measurements in solid lines. The lower graphs indicate a “1” when a fault is detected. In both graphs, the time is expressed in seconds, and the sample time is 1 s.

The fault begins at sample 900. For the  $r_2$  case, the fault  $f_{13}$  is detected from 903 s, and for the  $r_1$  case, this fault is detected from 915 s. Table 7.17 illustrates the diagnostics obtained either by considering, or not, the signs corresponding at the times in which each symptom appear.

Table 7.17: Diagnostics for the  $f_{13}$  fault scenario

Time (s)	903	915
Symptoms	$\text{sgn}(r_1) = 0$ $\text{sgn}(r_2) = +1$	$\text{sgn}(r_1) = +1$ $\text{sgn}(r_2) = +1$
Diagnosis	$f_1, f_2, f_3, f_4, f_8, f_{10}, f_{11}, f_{13}, f_{14}$	$f_2, f_3, f_8, f_{13}$
Diagnosis using signs	$f_3, f_4, f_8, f_{11}, f_{13}, f_{14}$	$f_3, f_8, f_{13}$



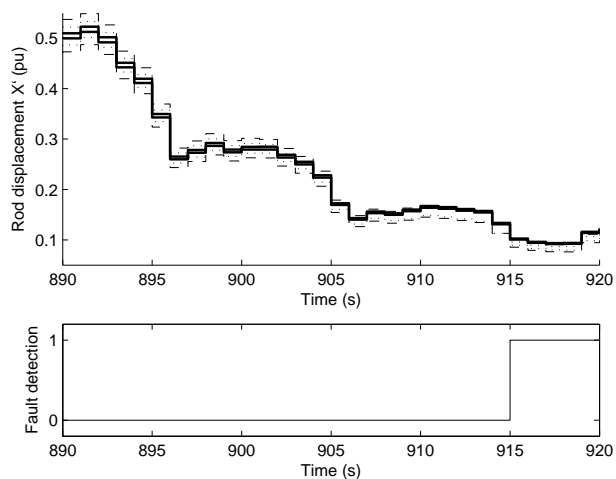


Figure 7.12: SQualTrack Fault Detection using  $r_1$  corresponding to fault  $f_{13}$  beginning at 900 s. The fault is detected from 915 s.

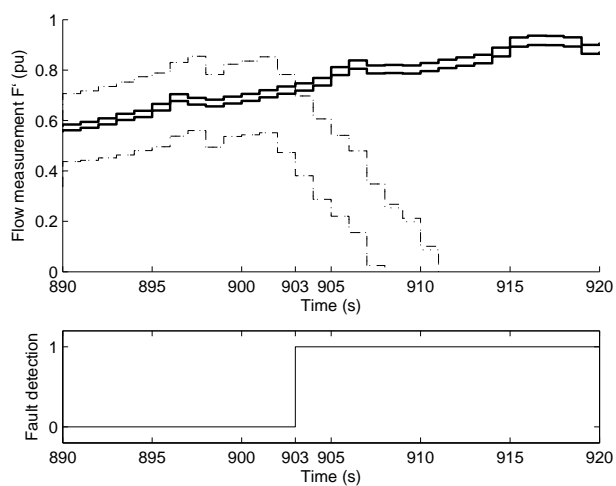


Figure 7.13: SQualTrack Fault Detection using  $r_2$  corresponding to fault  $f_{13}$  beginning at 900 s. The fault is detected from 903 s.

## 7.5 Alcoholic Fermentation Process

A well-known dynamical example of an alcoholic fermentation process is used to explain the proposed method for fault detection and quantitative fault isolation and identification. This process is presented in Li (2006); Li & Dahhou (2006, 2007); Li *et al.* (2006); Rajaraman (2005) and Kabbaj *et al.* (2001), for instance.

Regarding the approach proposed in Li & Dahhou (2007), the main contributions of this work are:

1. the isolation problem is based on parameters uncertainty refining instead of partitioning.
2. the isolation problem is stated as a Constraint Satisfaction Problem (CSP) and solved by means of consistency techniques. A sliding time window is used to reduce the computational effort.
3. interval calculations allow the proposed approach to be independent of the assumption (about the type of nonlinear systems) that the system dynamics is a monotonous function with respect to the considered parameters.

### 7.5.1 System Description

The fermentation consists in growing a population of microorganisms by feeding them appropriate nutrients or substrates, provided the environmental conditions are propitious Kabbaj *et al.* (2001).

The model obtained from the mass balance considerations is composed of the following differential equations:

$$\begin{cases} \frac{dC(t)}{dt} = \mu(t)C(t) - D(t)C(t) \\ \frac{dS(t)}{dt} = -\frac{1}{Y_{c/s}}\mu(t)C(t) + D(t)S_a - D(t)S(t) \\ \frac{dP(t)}{dt} = \frac{Y_{p/s}}{Y_{c/s}}\mu(t)C(t) + D(t)P(t) \end{cases} \quad (7.18)$$

where  $C(t)$ ,  $S(t)$  and  $P(t)$  represent respectively the biomass, substrate, and product concentrations in the bioreactor. The dilution rate  $D(t)$  is used as the control variable.  $S_a$  represents the substrate concentration in the feeding.  $Y_{c/s}$  and

## 7.5 Alcoholic Fermentation Process

---

$Y_{p/s}$  are the yield coefficients and it is assumed that they are known and constant. The measurable state is the substrate concentration  $S(t)$ .  $\mu(t)$  represents the growth rate of the biomass, and it is a nonlinear function of the variable  $S(t)$  described by

$$\mu(t) = \mu_m \frac{S(t)}{K_s + S(t)} \quad (7.19)$$

where  $\mu_m$  is the maximum growth rate and  $K_s$  is the saturation constant.

Faults are modeled as a single parameter change in the process parameters  $\mu_m$  and  $K_s$ .

The interval method uses discrete-time models. In this case a discretization is obtained by using a first order approximation:

$$\mathbf{x}(t + T_s) \simeq \mathbf{x}(t) + T_s \mathbf{g}(\mathbf{x}(t), \mathbf{u}(t), \boldsymbol{\theta}), \quad (7.20)$$

where the sample time,  $T_s$ , is equal to 3 minutes.

Thus, from Eq. 7.18, the following discrete-time model can be obtained:

$$\begin{aligned} \hat{C}(k+1) &= \hat{C}(k) + T_s(\mu(k)\hat{C}(k) - \tilde{D}(k)\hat{C}(k)) + w_1(k) \\ \hat{S}(k+1) &= \hat{S}(k) - T_s\left(\frac{\mu(k)}{Y_{c/s}}\hat{C}(k) - \tilde{D}(k)(S_a - \hat{S}(k))\right) + w_2(k) \\ \hat{P}(k+1) &= \hat{P}(k) + T_s\left(\frac{Y_{p/s}}{Y_{c/s}}\mu(k)\hat{C}(k) + \tilde{D}(k)\hat{P}(k)\right) + w_3(k) \\ \tilde{S}(k) &= \hat{S}(k) + v(k) \end{aligned} \quad (7.21)$$

where  $w_i(k)$  is the perturbation vector at time  $k$ , and it takes into account, for example, an error due to the discretization procedure.  $v(k)$  is the measurement noise of the interval measurement  $\tilde{S}(k)$ .

In the simulation,  $D(t)$  is selected as a rectangular wave varying between 0.1 and 0.27 h<sup>-1</sup> with a period of 30 hours. The sample time,  $T_s$ , is equal to 3 minutes. The feasible ranges of parameter variation, i.e. experimental considerations in practice are given by  $\mu_m \in [0.2, 0.53]$  h<sup>-1</sup> and  $K_s \in [0.5, 5.1]$  g/l.

The nominal values of model parameters used as well as the yield coefficients are obtained from real applications and are given by Li & Dahhou (2007) (see Table 7.18).

Table 7.18: Parameters and yield coefficients.

Nominal Values
$\mu_m = 0.38 \text{ h}^{-1}$
$K_s = 5 \text{ g/l}$
$Y_{c/s} = 0.07$
$Y_{p/s} = 0.44$
$S_a = 100 \text{ g/l}$

### 7.5.2 Fault Detection Results

Considered faults are modeled as a single parameter change in the process parameters  $\mu_m$  and  $K_s$ . Therefore two faulty scenarios are considered.

The fault detection results (see Table 7.19) are obtained by using BC4 consistency technique [Granvilliers \(2004\)](#) and a sliding window length equal to 100 samples (5 h). All times displayed are relative to the start of the experiment.

Table 7.19: Fault detection results

Scenario	Faulty parameter	Nominal range	Faulty value	Fault occurrence time (h)	Detection time (h)
(i)	$\mu_m$	[0.36, 0.41]	0.3	70	70.05
(ii)	$K_s$	[4.90, 5.10]	3.1	70	70.35

In Figs. 7.14 and 7.15, obtained results for both fault scenarios are shown. “1” indicates there is a fault and a “0” means there is not a fault or one could not be detected. As shown in these figures, there are no false alarms.

### 7.5.3 Fault Isolation Results

Once the fault is detected, the fault isolation algorithm starts. Since the parameters of this case study are  $\mu_m$ , and  $K_s$ , two CSP must be solved. For the first, the set of initial domains of the parameters is:  $(\mu_m^p, K_s^0)$ , and for the last one  $(\mu_m^0, K_s^p)$ . When no consistent region is found in the feasible range of parameter

## 7.5 Alcoholic Fermentation Process

---

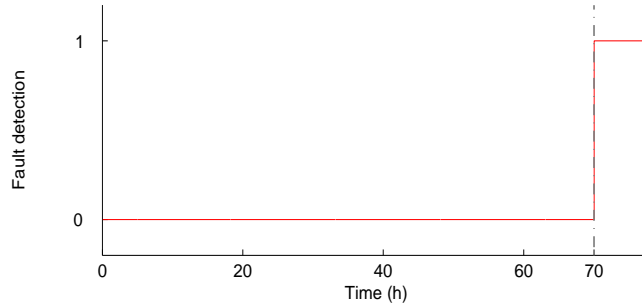


Figure 7.14: Fault detection. Scenario (i), fault is on parameter  $\mu_m$ .

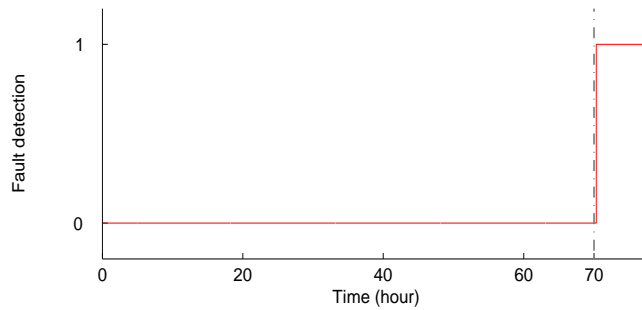


Figure 7.15: Fault detection. Scenario (ii), fault is on parameter  $K_s$ .

variation, a fault associated with a value change of the refined parameter can be discarded. Obtained results for fault isolation in both scenarios are summarized in Table 7.20.

Table 7.20: Fault isolation results

Scenario	Faulty parameter	Feasible range	Faulty value	Estimated range	Isolation time (h)
(i)	$\mu_m$	[0.20, 0.53]	0.3	[0.282, 0.314]	70.70
(ii)	$K_s$	[0.50, 5.10]	3.1	[2.890, 3.250]	75.20

Notice that in both scenarios the estimated range of the faulty parameter includes the considered faulty value.

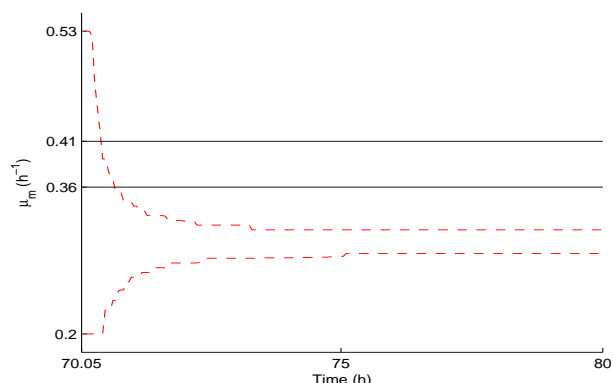
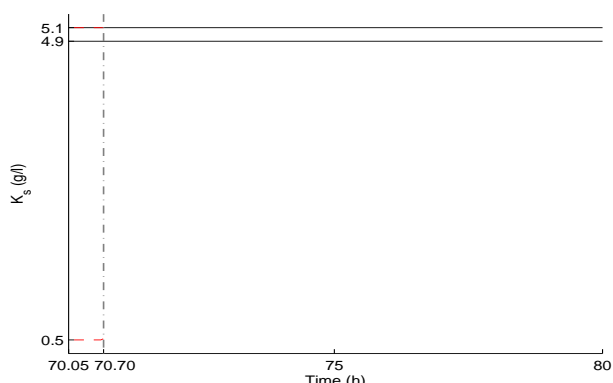
In Table 7.21 and Table 7.22, the results of the isolation test are presented in a more detailed way.

## 7.5 Alcoholic Fermentation Process

- Faulty parameter  $\mu_m$

Since there is no consistent region of  $K_s$  in its feasible range of variation, the fault is not in the parameter  $K_s$  and it can be discarded at time 70.70 h (See Table 7.21).

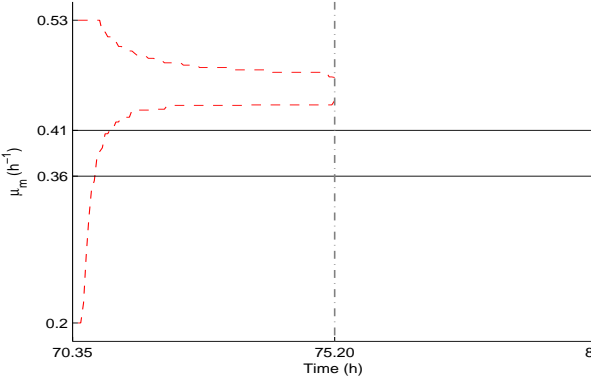
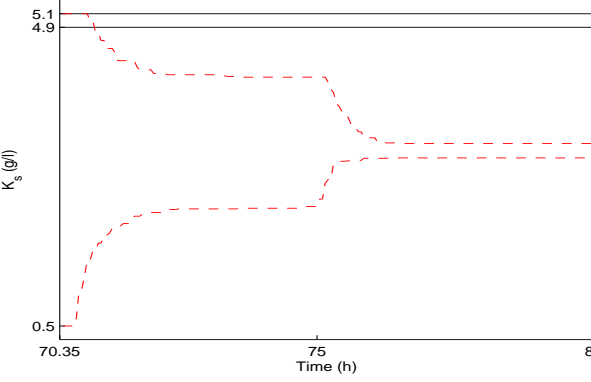
Table 7.21: Scenario(i). Fault isolation results

Test	Isolation Results
Test 1: Using the nominal range of $K_s$ and refining the feasible range of $\mu_m$	 <p style="text-align: center;">Consistent values for <math>\mu_m</math></p>
Test 2: Using the nominal range of $\mu_m$ and refining the feasible range of $K_s$	 <p style="text-align: center;">Consistent values for <math>K_s</math></p>

- Faulty parameter  $K_s$

Since there is no consistent region of  $\mu_m$  in its feasible range of variation, the fault is not in the parameter  $\mu_m$  and it can be discarded at time 75.20 h (See Table 7.22).

Table 7.22: Scenario(ii). Fault isolation results

Test	Isolation Results
<p>Test 1: Using the nominal range of <math>K_s</math> and refining the feasible range of <math>\mu_m</math></p>	 <p style="text-align: center;">Consistent values for <math>\mu_m</math></p>
<p>Test 2: Using the nominal range of <math>\mu_m</math> and refining the feasible range of <math>K_s</math></p>	 <p style="text-align: center;">Consistent values for <math>K_s</math></p>

### 7.5.4 Isolation Time

Even taking into account uncertainty in measurements, parameters, and model errors, the isolation time in both scenarios obtained by means of the proposed approach, is of the same order of magnitude than the ones found in [Li & Dahhou \(2007\)](#), which uses an interval approach, non-based on interval calculations, for similar faulty scenarios. It could be said that the two approaches isolation speeds are comparable, with the advantage that the method proposed in this thesis does not require the monotonous condition of the dynamic system with respect to the considered parameters.

# Chapter 8

## Conclusions and Future Work

This thesis looks into the detection, isolation, and identification of faults under uncertain conditions. The research starting point was SQualTrack [Armengol \*et al.\* \(2009b\)](#), a software package for fault detection based on interval models. By analyzing the capabilities and limitations of this software, four main aspects were considered to bound the research problem of this thesis:

- Use interval models to handle the uncertainty associated with the process itself and with the measurements.
- Overcome the limitations of SQualTrack discussed in Chapter 4.
- Propose solutions based on interval models for fault isolation and identification.
- Include approaches from FDI and DX compatible with interval models in the solution of the problem.

Summarizing the previous aspects, the main objective of this thesis became: to propose a methodology for fault detection, isolation and identification based on interval models. This methodology includes both off-line and online stages within the supervision process.

Following the scheme of the diagnosis system depicted in Fig. 8.1, the main objective is achieved as follows.



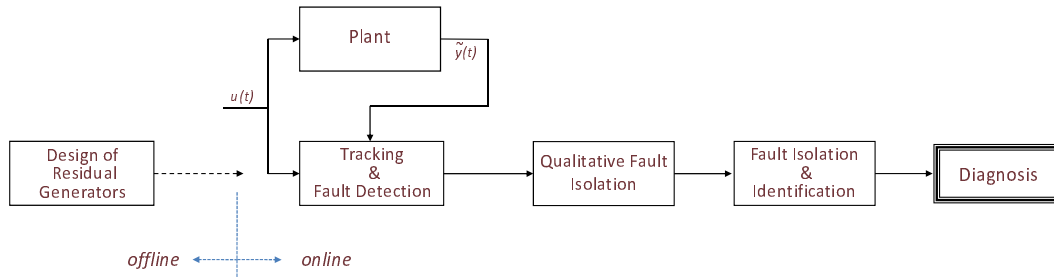


Figure 8.1: Architecture of the diagnosis system.

One of the main contributions of Chapter 3 is the proposed algorithm to find all MSO sets in a structural model of the system. MSO sets are useful for designing diagnostic tests, by finding in an efficient way all the subsystems that could be diagnosed in the system. The algorithm is an improvement of the one presented in Gelso *et al.* (2008c).

In Chapter 3, an algorithm that analyzes the input to output active structural isolation of a system is also developed. By means of this technique, fault detectability and isolability properties can be better, in some cases, than the ones obtained using a passive approach, for instance one that includes the set of all MSO sets and the corresponding fault signature matrix. Based on this, Chapter 3 shows that both the passive and the active techniques can be used in a complementary way to improve the detectability and isolability properties. Obtained results regarding active structural isolation appear in Gelso & Blanke (2009).

For the tracking and fault detection stage, two improvements are proposed in Chapter 4 to increase the fault detection performance. The former refers to the dynamic refinement of the parameters space, and the latter, to the pruning of the measurements space. These improvements are described in Gelso *et al.* (2007a).

Because of some limitations when implementing ARR in SQualTrack, for example ARRs are restricted to be expressed as a single equation, another approach based on interval-based consistency techniques (ICTs) is proposed. By using the ICTs, ARRs can be described in a state space representation with multiple state variables. Gelso *et al.* (2007c, 2008e) include the work based on ICTs used for fault detection and its comparison with SQualTrack.

---

In Chapter 4, a technique based on a robust observer and a statistical fault detector are empirically compared to an ICT. In general, comparing the performance parameters of the fault detection, both techniques yielded similar fault detection results for a proper combination of tuning parameters. In both techniques, a compromise between false alarms, missed detections and detection delays can be made varying the noise confidence interval. The empirical comparison is presented in [Gelso \*et al.\* \(2008a\)](#).

In Chapter 5, a diagnosis reasoning in which the signs of the partial derivatives are derived from analytical redundancy relations, is proposed to improve the task of diagnosis. The advantages of an interval tool, the SQualTrack, can be exploited to evaluate the consistency between a model and a system for fault diagnosis, and to give the qualitative deviations of the residuals. This method is presented in [Calderón-Espinoza \*et al.\* \(2007\)](#); [Gelso \*et al.\* \(2007b\)](#).

Chapter 5 also presents a method implemented in the TRANSCEND methodology that generates qualitative fault signatures when a bond graph model of the system is available. This technique is briefly compared with the technique based on partial derivatives. The technique based on bond graphs is extended to the analysis of symbol generation for faults with a discontinuous change in a measurement. Furthermore, a way to distinguish signatures with a discontinuous and non-discontinuous effect, based on discrete wavelet transforms, is proposed.

In Chapter 6, a quantitative fault isolation and identification scheme based on ICTs to refine the fault hypothesis test and to estimate the fault magnitude, is proposed. The use of overconstrained subsystems of the model leads the approach to tackle the fault identification problem in smaller subproblems which can reduce the complexity of the problem. CSPs are used also to represent the fault identification problem, and hence, to calculate the fault magnitude. Results obtained using this proposed approach are presented in [Gelso \*et al.\* \(2008b,d\)](#).

In order to illustrate the effectiveness of the methods and the theory developed in this thesis, several application examples are presented in Chapter 7. These applications include:

1. Positioning control system of an offshore vessel.
2. Electrical distribution systems.

- 
3. Automotive engine.
  4. DAMADICS Benchmark.
  5. Alcoholic Fermentation Process.

## Future Work

Regarding the future work, there are open fields on the subject of fault detection, isolation and identification, that could be tackled. And, specifically talking about the work done in this thesis, it could be extended in several directions. In the following, these aspects are briefly outlined.

Active injection of test signals on system inputs can considerably enhance fault isolability. In this sense, active isolation from a structural point of view, which is a rather new topic, was approached in this thesis. A further study in this line of work can be, for example, to analyze and to develop algorithms to carry out the input to output active structural isolation.

In this thesis proposed algorithms to find all MSO sets in a structural model of the system, were applied successfully to several application examples with different complexity level. However, the efficiency of them can be significantly improved by performing an optimization of the code.

The empirical comparisons between the fault detection techniques presented in this thesis allowed to see some of their capabilities and limitations. A theoretical analysis of the false alarm rate and missed alarm rate can be performed to link the tuning parameters of the methods. In future an in depth formal comparison between different fault detection techniques could originate new research works. For instance, the comparison between interval- and statistical-based methods for fault detection presented in this work, could be extended to other fault detection methods like the ones described in [Baseville & Nikiforov \(1993\)](#); [Manders & Biswas \(2003\)](#); [Niemann \*et al.\* \(1999\)](#).

The fault isolation stage could be improved by considering other sources of information, like the order of the symptom appearance of a given ARR with respect to the others. An alternative to perform this improvement is by using

---

information derived from bond graphs models. In [Daigle \(2008\)](#), this method is shown to be useful to tackle multiple faults.

Future work also, will consist of analyzing the capability of the Temporal Causal Graph to analyze the sign of the symptoms when sensor faults occur. In particular, an important issue is how to obtain the theoretical signature of these faults when observer-based and prediction-based approaches are used to track the nominal behavior of the system.

Regarding the quantitative fault isolation and identification stage, other interval-based techniques could be explored to perform the faulty parameters estimation. Possible approaches to perform this task are the set-membership identification methods based on polytopes or zonotopes.

A feasibility study in order to extend the methodology of fault detection, isolation and identification, to continuous time models, could be performed. This research could consider interval techniques such as the ones proposed by [Lin & Statherr \(2007, 2008\)](#).



# References

- Acosta, G., Alonso, C. & Pulido, B. (2002). Basic tasks for knowledge-based supervision in process control. *Engineering Applications of Artificial Intelligence*, **14**, 441–455. [54](#)
- Acosta, G.G., Verucchi, C.J. & Gelso, E.R. (2006). A current monitoring system for diagnosing electrical failures in induction motors. *Mechanical Systems and Signal Processing*, **20**, 953–965. [15](#)
- Andersson, P. (2005). *Air Charge Estimation in Turbocharged Spark Ignition Engines*. Ph.D. thesis, Linköping University. [146](#)
- Armengol, J. (1999). *Application of Modal Interval Analysis to the simulation of the behavior of dynamic systems with uncertain parameters*. Phd thesis, Universitat de Girona, Girona, Catalonia, Spain. [2](#)
- Armengol, J., Travé-Massuyès, L., Vehí, J. & de la Rosa, J.L. (2000). A survey on interval model simulators and their properties related to fault detection. *Annual Reviews in Control*, **24**, 31–39. [25](#)
- Armengol, J., Vehí, J., Travé-Massuyès, L. & Sainz, M.Á. (2001a). Application of modal intervals to the generation of error-bounded envelopes. *Reliable Computing*, **7**, 171–185. [60](#)
- Armengol, J., Vehí, J., Travé-Massuyès, L. & Sainz, M.Á. (2001b). Application of multiple sliding time windows to fault detection based on interval models. In S. McIlraith & D. Theseider Dupré, eds., *12th International Workshop on Principles of Diagnosis DX 2001. San Sicario, Italy*, 9–16. [64](#)

- Armengol, J., Vehí, J., Sainz, M.Á. & Herrero, P. (2003). Fault detection in a pilot plant using interval models and multiple sliding time windows. In *5th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes SAFEPROCESS 2003. Washington, D.C., U.S.A.*. 27, 60, 62
- Armengol, J., Bregón, A., Escobet, T., Gelso, E., Krysander, M., Nyberg, M., Olive, X., Pulido, B. & Travé-Massuyès, L. (2009a). Minimal structurally overdetermined sets for residual generation: A comparison of alternative approaches. In *7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes SAFEPROCESS*, Accepted, Barcelona, Spain. 58
- Armengol, J., Vehí, J., Sainz, M.Á., Herrero, P. & Gelso, E.R. (2009b). SQual-Track: a tool for robust fault detection. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, **39**, 475–488. 27, 60, 62, 171
- Balakrishnan, K. & Honavar, V. (1998). Intelligent diagnosis systems. *Journal of Intelligent Systems*, **8**, 239–290. 12
- Bartyś, M. & de las Heras, S. (2003). Actuator simulation for the DAMADICS benchmark actuator system. In *5th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes SAFEPROCESS*, 963–968, Washington. 157
- Bartyś, M. & Syfert, M. (2002). Using DAMADICS actuator benchmark library (DABlib). *In manuscript on: <http://diag.mchtr.pw.edu.pl/damadics/>*. 157
- Bartyś, M., Patton, R., Syfert, M., de las Heras, S. & Quevedo, J. (2006). Introduction to the DAMADICS actuator FDI benchmark study. *Control Engineering Practice*, **14**, 577–596. 156
- Baseville, M. & Nikiforov, I. (1993). *Detection of abrupt changes: theory and application*. Prentice Hall. 174
- Basseville, M. (1988). Detecting changes in signals and systems - a survey. *Automatica*, **24**, 309–326. 22

- 
- Benhamou, F., Goualard, F., Granvilliers, L. & Puget, J.F. (1999). Revising hull and box consistency. In *Proceedings of the International Conference on Logic Programming*, 230–244, Las Cruces, NM. [74](#)
- Berleant, D. & Kuipers, B. (1992). Qualitative-numeric simulation with Q3. In B. Faltings & P. Struss, eds., *Recent advances in qualitative physics*, 3–16, MIT Press. [27](#)
- Bishop, R. (2002). *The Mechatronics Handbook*. CRC Press. [121](#)
- Biswas, G., Simon, G., Mahadevan, N., Narasimhan, S., Ramirez, J. & Karsai, G. (2003). A robust method for hybrid diagnosis of complex systems. In *5th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes SAFEPROCESS*, 1125–1131. [85](#), [122](#)
- Biteus, J. (2007). *Fault Isolation in Distributed Embedded Systems*. Ph.D. thesis, Linköping University. [18](#)
- Blanke, M. & Lorentzen, T. (2006). Satool - a software tool for structural analysis of complex automation systems. In *6th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes*, Beijing, China. [30](#), [43](#), [46](#), [57](#)
- Blanke, M. & Staroswiecki, M. (2006). Structural design of systems with safe behavior under single and multiple faults. In *6th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes SAFEPROCESS*, 511–516, Beijing, China. [50](#), [51](#)
- Blanke, M., Kinnaert, M., Lunze, J. & Staroswiecki, M. (2006). *Diagnosis and Fault-Tolerant Control*. Springer, 2nd edn. [16](#), [20](#), [29](#), [30](#), [32](#), [34](#), [35](#), [36](#), [46](#), [48](#)
- Bondy, J. & Murty, U. (1976). *Graph Theory With Applications*. Elsevier Science Ltd. [35](#)
- Bousson, K. & Travé-Massuyès, L. (1994). Putting more numbers in the qualitative simulator CA-EN. *2nd International Conference on Intelligent Systems Engineering ISE 94. Hamburg-Harburg, Germany*, 62–69. [27](#)



- Calderón, G., Armengol, J., Sainz, M.Á. & Herrero, P. (2005). Combining interval and qualitative reasoning for fault diagnosis. In *16th IFAC World Congress*, Praga. [102](#)
- Calderón-Espinoza, G., Armengol, J., Vehí, J. & Gelso, E. (2007). Dynamic diagnosis based on interval analytical redundancy relations and signs of the symptoms. *AI Communications*, **20**, 39–47. [173](#)
- Campbell, S., Horton, K., Nikoukha, R. & Delebecque, F. (2000). Rapid model selection and the separability index. In *4th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes SAFEPROCESS*, 1187–1192, Budapest, Hungary. [50](#)
- Castillo, S.M. (2007). *Estimación de parámetros intervalares para la detección de fallos*. Diploma of advanced studies - DEA, Universitat de Girona, Girona. [66](#)
- Castillo, S.M., Gelso, E.R. & Armengol, J. (2007). Modelos con incertidumbre: Identificación de parámetros intervalares y simulación usando cuantificadores. In *II Congreso Español de Informatica (CEDI'07), I Simposio en Modelado y Simulación de Sistemas Dinámicos*, Zaragoza, Spain. [66](#)
- Ceballos, R., Gómez, M., Gasca, R. & Pozo, S. (2004). Determination of possible minimal conflict sets using components clusters and Gröbner bases. In *15th International Workshop on Principles of Diagnosis DX*, 21–26. [32](#)
- Chang, I., Yu, C. & C.T., L. (1994). Model-based approach for fault diagnosis: Part I. principles of deep model algorithm. *Ind. Eng. Chem. Res.*, **33**, 1542–1555. [102](#), [104](#)
- Chantler, M., Coghill, G., Shen, Q. & Leitch, R. (1998). Selecting tools and techniques for model-based diagnosis. *Artificial Intelligence in Engineering*, **12**, 81–98. [16](#)
- Chen, J. & Patton, R. (1998). *Robust model-based fault diagnosis for dynamic systems*. Kluwer. [20](#), [22](#), [23](#), [24](#)

- Chow, E.Y. & Willsky, A.S. (1980). Issues in the development of a general design algorithm for reliable failure detection. In *19th IEEE Conference on Decision and Control*, vol. 19, 1006–1012, Albuquerque, NM. [21](#)
- Chow, E.Y. & Willsky, A.S. (1984). Analytical redundancy and the design of robust failure detection systems. *IEEE Transactions on Automatic Control*, **29**, 603–614. [21](#)
- Christophe, C., Cocquempot, V. & Jiang, B. (2002). Link between high gain observer-based residual and parity space one. In *American Control Conference, 2002*, vol. 3, 2100–2105. [23](#)
- Chui, C. (1997). *Wavelets: A Mathematical Tool for Signal Analysis*. SIAM. [123](#), [124](#)
- Coghill, G. (1996). *Mycroft: a framework for constraint based fuzzy qualitative reasoning*. Ph.D. thesis, Heriot-Watt University, Edinburgh, Scotland. [26](#)
- Collavizza, H., Delobel, F. & Rueher, M. (1999). Comparing partial consistencies. *Reliable Computing*, **5**, 213–228. [74](#), [75](#)
- Combastel, C., Zhang, Q. & Lalami, A. (2008). Fault diagnosis based on the enclosure of parameters estimated with an adaptive observer. In *17th IFAC World Congress*, 7314–7319, Seoul, Korea. [27](#), [61](#)
- Console, L., Correndo, G. & Picardi, C. (2003). Deriving qualitative deviations from Matlab models. In *14th International Workshop on Principles of Diagnosis DX*. [102](#)
- Cordier, M.O., Dague, P., Lévy, F., Montmain, J., Staroswiecki, M. & Travé-Massuyès, L. (2004). Conflicts versus analytical redundancy relations: a comparative analysis of the model based diagnosis approach from the artificial intelligence and automatic control perspectives. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, **34**, 2163–2177. [17](#), [30](#), [48](#), [49](#), [106](#)
- Cormen, T., Leiserson, C., Rivest, R. & Stein, C. (2003). *Introduction to Algorithms*. MIT Press, 2nd edn. [51](#)

- 
- Cruz, J. & Barahona, P. (2002). Maintaining global hull consistency with local search for continuous CSPs. In *First International Workshop on Global Constraint Optimization and Constraint Satisfaction, COCOS 2002*, 178–193. [74](#), [75](#)
- Daigle, M. (2008). *A qualitative event-based approach to fault diagnosis of hybrid systems*. Ph.D. thesis, Vanderbilt University, Nashville (USA). [19](#), [104](#), [115](#), [117](#), [122](#), [175](#)
- Daigle, M., Koutsoukos, X. & Biswas, G. (2006). Multiple fault diagnosis in complex physical systems. In *17th International Workshop on Principles of Diagnosis DX*, 69–76. [19](#)
- de Kleer, J. (1986). An assumption-based TMS. *Artificial Intelligence*, **28**, 127–162. [19](#)
- de Kleer, J. & Kurien, J. (2003). Fundamentals of model-based diagnosis. In *14th International Workshop Principles of Diagnosis DX*. [17](#), [19](#), [20](#)
- de Kleer, J. & Williams, B. (1987). Diagnosing multiple faults. *Artificial Intelligence*, **32**, 97–130. [17](#), [18](#), [19](#)
- de Kleer, J., Mackworth, A. & Reiter, R. (1992). Characterizing diagnoses and systems. *Artificial Intelligence*, **56**, 197–222. [17](#)
- Diestel, R. (2000). *Graph Theory*. Springer-Verlag, 2nd edn. [35](#)
- DNV (2008). *Rules for Classification of Ships*. Det Norske Veritas, Høvik, Norway. [138](#)
- Dressler, O. & Struss, P. (1996). The consistency-based approach to automated diagnosis of devices. 269–314. [17](#), [19](#), [20](#)
- Dulmage, A. & Mendelsohn, N. (1958). Coverings of bipartite graphs. *Canadian Journal of Mathematics*, **10**, 517–534. [41](#)

- Dulmage, A. & Mendelsohn, N. (1963). Two algorithms for bipartite graphs. *Journal of the Society for Industrial and Applied Mathematics*, **11**, 183–194. [41](#)
- Düştögör, D. (2005). *Structural Analysis for Fault Detection and Identification: Algorithmic Issues*. Ph.D. thesis, Automatic, Computer and System Eng., Université des Sciences et Technologies de Lille, Lille (France). [31](#), [40](#)
- Düştögör, D., Cocquempot, V. & Staroswiecki, M. (2004). Structural analysis for residual generation: towards implementation. In *IEEE International Conference on Control Applications*, vol. 2, 1217–1222. [31](#)
- Düştögör, D., Frisk, E., Cocquempot, V., Krysander, M. & Staroswiecki, M. (2006). Structural analysis of fault isolability in the DAMADICS benchmark. *Control Engineering Practice*, **14**, 597–608. [33](#), [45](#), [50](#)
- EA Internacional (2008). Ecosimpro, modelling and simulation software, user manual version 4.4. <http://www.ecosimpro.com>. [147](#)
- Fagarasan, I., Ploix, S. & Gentil, S. (2004). Causal fault detection and isolation based on a set-membership approach. *Automatica*, **40**, 2099–2110. [27](#), [60](#)
- Frank, P. (1990). Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy - a survey and some new results. *Automatica*, **26**, 459–474. [9](#)
- Frisk, E. (2000). Residual generator design for non-linear, polynomial systems - A gröbner basis approach. In *4th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes SAFEPROCESS*, 979–984, Budapest, Hungary. [32](#)
- Gardenyes, E., Sainz, M.Á., Jorba, L., Calm, R., Estela, R., Mielgo, H. & Trepát, A. (2001). Modal intervals. *Reliable Computing*, **7**, 77–111. [60](#)
- Gasca, R., Ortega, J. & Toro, M. (2002). A framework for semiqualitative reasoning in engineering applications. *Applied Artificial Intelligence*, **16**, 173–197. [27](#)

- Gasca, R.M., Ortega, J.A. & Toro, M. (1996). Simulación semicualitativa. *Jornades Hispano-Franceses de Sistemes Intel·ligents i Control Avançat*. Barcelona, Catalonia, Spain. 27
- Gawthrop, P. & Bevan, G. (2007). Bond-graph modeling. *IEEE Control Systems Magazine*, 27, 24–45. 113, 114
- Gelb, A. (1996). *Applied Optimal Estimation*. MIT Press. 85
- Gelso, E.R. & Blanke, M. (2009). Structural analysis extended with active fault isolation – methods and algorithms. In *7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes SAFEPROCESS*, Accepted, Barcelona, Spain. 172
- Gelso, E.R., Castillo, S.M. & Armengol, J. (2007a). Construyendo una plataforma para detección y diagnosis de sistemas dinámicos basada en análisis intervalar modal y redundancia analítica. In *IX Jornadas de ARCA. Sistemas Cualitativos y Diagnosis*, 15–21, Lloret de Mar, Spain. 172
- Gelso, E.R., Castillo, S.M. & Armengol, J. (2007b). Diagnosis based on interval analytical redundancy relations and signs of the symptoms. In *IFAC International Workshop on Intelligent Manufacturing Systems IMS'07*, Alicante, Spain. 173
- Gelso, E.R., Castillo, S.M. & Armengol, J. (2007c). Robust fault detection using consistency techniques for uncertainty handling. In *IEEE International Symposium on Intelligent Signal Processing, WISP 2007*, 77–82, Alcalá de Henares, Spain. 27, 61, 172
- Gelso, E.R., Biswas, G., Castillo, S.M. & Armengol, J. (2008a). A comparison of two methods for fault detection: a statistical decision, and an interval-based approach. In *19th International Workshop on Principles of Diagnosis DX*, 261–268, Blue Mountains, Australia. 173
- Gelso, E.R., Castillo, S.M. & Armengol, J. (2008b). Fault diagnosis by refining the parameter uncertainty space of nonlinear dynamic systems. In *American Control Conference (ACC08)*, Seattle, USA. 173

- Gelso, E.R., Castillo, S.M. & Armengol, J. (2008c). *Frontiers in Artificial Intelligence and Applications - Artificial Intelligence Research and Development (CCIA'08)*, chap. An algorithm based on structural analysis for model-based fault diagnosis, 138–147. IOS Press. 172
- Gelso, E.R., Castillo, S.M. & Armengol, J. (2008d). *Frontiers in Artificial Intelligence and Applications - Artificial Intelligence Research and Development (CCIA'08)*, chap. An interval-based approach for fault isolation and identification in continuous dynamic systems, 421–429. IOS Press. 173
- Gelso, E.R., Frisk, E. & Armengol, J. (2008e). Robust fault detection using consistency techniques with application to an automotive engine. In *17th IFAC World Congress*, 5512–5517, Seoul, Korea. 172
- Gentil, S., Montmain, J. & Combastel, C. (2004). Combining FDI and AI approaches within causal-model-based diagnosis. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, **34**, 2207–2221. 17
- Gertler, J., Costin, M., Fang, X., Hira, R., Kowalalczuk, Z., Kunwer, M. & Monajemy, R. (1995). Model based diagnosis for automotive engines - algorithm development and testing on a production vehicle. *IEEE Trans. on Control Systems Technology*, **3**, 61–69. 146
- Gertler, J.J. (1998). *Fault Detection and Diagnosis in Engineering Systems*. Marcel Dekker. 11, 14, 20, 22, 104
- Granvilliers, L. (2004). Realpaver user's manual, version 0.4. <http://www.sciences.univ-nantes.fr/info/perso/permanents/granvil/realpaver/>. 82, 92, 151, 167
- Granvilliers, L. & Benhamou, F. (2006). Algorithm 852: Realpaver, an interval solver using constraint satisfaction techniques. *ACM Trans. Math. Softw.*, **32**, 138–156. 4, 76
- Granvilliers, L., Goualard, F. & Benhamou, F. (1999). Box consistency through weak box consistency. In *Proceedings of the 11th IEEE International Conference on Tools with Artificial Intelligence (ICTAI99)*, 373–380. 61

- Guerra, P., Bravo, J.M., Ingimundarson, A., Puig, V. & Álamo, T. (2006). Robust fault detection based on zonotope-based set-membership parameter consistency test. In *6th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes SAFEPROCESS*. 61
- Guerra, P., Puig, V. & Ingimundarson, A. (2007). Passive robust fault detection: Inverse vs direct image tests using zonotopes. In *18th International Workshop on Principles of Diagnosis DX*, 122–129, Nashville, USA. 61
- Hamscher, W., Console, L. & de Kleer, J. (1992). *Readings in Model-Based Diagnosis*. Morgan Kaufman, San Francisco, CA, USA. 17
- Hansen, E. (1992). *Global optimization using interval analysis*. Marcel Dekker. 26
- Herrero, P. (2006). *Quantified Real Constraint Solving Using Modal Intervals with Applications to Control*. Ph.D. thesis, Universitat de Girona, Girona, Spain. 64, 65, 66
- Ingimundarson, A., Bravo, J.M., Puig, V. & Álamo, T. (2005). Robust fault diagnosis using parallelotope-based set-membership consistency tests. In *Proceedings of the 44th IEEE Conference on Decision and Control*, 993–998, Seville, Spain. 27, 61
- Ingimundarson, A., Bravo, J.M., V., P., Álamo, T. & Guerra, P. (2009). Robust fault detection using zonotope-based set-membership consistency test. *International Journal of Adaptive Control and Signal Processing*, To appear. 27, 61
- Isermann, R. (2005). Model-based fault-detection and diagnosis - status and applications. *Annual Reviews in Control*, 29, 71–85. 10
- Isermann, R. (2006). *Fault-Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance*. Springer Verlag. 10, 20, 22
- Isermann, R. & Ballé, P. (1997). Trends in the application of model-based fault detection and diagnosis of technical processes. *Control Engineering Practice*, 5, 707–719. 1, 10

- Izadi-Zamanabadi, R. (2002). Structural analysis approach to fault diagnosis with application to fixed-wing aircraft motion. *American Control Conference, 2002*, **5**, 3949–3954. [30](#)
- Janati-Idrissi, H., Adrot, O., Ragot, J. & Kratz, F. (2002). Fault detection of uncertain models using polytope projection. In *15th IFAC World Congress*. [27](#), [61](#)
- Jaulin, L. (2002). Consistency techniques for the localization of a satellite. In *First International Workshop on Global Constraint Optimization and Constraint Satisfaction, COCOS 2002*, 157–170. [72](#), [74](#)
- Kabbaj, N., Polit, M., Dahhou, B. & Roux, G. (2001). Adaptive observers based fault detection and isolation for an alcoholic fermentation process. In *8th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA'2001)*, Antibes - Juan les Pins (France). [165](#)
- Karim, J., Jauberthie, C. & Combacau, M. (2008). Model-based fault detection method using interval analysis: Application to an aeronautic test bench. In *19th International Workshop on Principles of Diagnosis DX*, 269–273. [27](#)
- Karnopp, D., Margolis, D. & Rosenberg, R. (2000). *System dynamics: modeling and simulation of mechatronic systems*. John Wiley & Sons, Inc., 3rd edn. [113](#)
- Kaucher, E. (1979). Interval analysis in the extended interval space IR. *Computing Supplementum*, **2**, 33–49. [60](#)
- Kay, H. (1996). *Refining imprecise models and their behaviours*. Ph.D. thesis, University of Texas at Austin. [27](#)
- Khosravi, A., Melendez, J., Colomer, J. & Sanchez, J. (2007). A hybrid method for sag source location in power networks. In *9th International Conference on Electrical Power Quality and Utilisation EPQU'2007*, Barcelona, Spain. [15](#)
- Khosravi, A., Zapatero, M., Melendez, J. & Colomer (2008). Classification of voltage sags based on MPCA and CBR. In *17th IFAC World Congress*, 5529–5534, Seoul, Korea. [15](#)



- Kimmich, F., Schwarte, A. & Isermann, R. (2005). Fault detection for modern diesel engines using signal- and process model-based methods. *Control Engineering Practice*, **13**, 189–203. [146](#)
- Knüppel, T. (2008). *Structural Analysis for Fault Detection and Isolation in Electrical Distribution Systems*. Master’s thesis, Department of Electrical Engineering, Technical University of Denmark. [142](#), [143](#), [144](#), [145](#)
- Kościelny, J.M., Bartyś, M., Rzepiejewski, P. & Sá da Costa, J. (2006). Actuator fault distinguishability study for the DAMADICS benchmark problem. *Control Engineering Practice*, **14**, 645–652. [103](#), [104](#)
- Krysander, M., Åslund, J. & Nyberg, M. (2008). An efficient algorithm for finding minimal over-constrained sub-systems for model-based diagnosis. *IEEE Trans. on Systems, Man, and Cybernetics – Part A: Systems and Humans*, **38**. [30](#), [31](#), [33](#), [34](#), [43](#)
- Kuipers, B. (1990). *Readings in qualitative reasoning about physical systems*, chap. Qualitative simulation, 236–260. Morgan Kaufmann Publishers Inc. [26](#)
- Kuipers, B. (1994). *Qualitative reasoning. Modeling and simulation with incomplete knowledge*. MIT Press. [26](#)
- Kuipers, B. & Berleant, D. (1988). Using incomplete quantitative knowledge in qualitative reasoning. In *Proc. of the Sixth National Conference on Artificial Intelligence*, 324–329, Saint Paul (USA). [27](#)
- Laursen, M., Blanke, M. & Düstegör, D. (2008). Fault diagnosis in a water for injection system using enhanced structural isolation. *International Journal of Applied Mathematics and Computer Science*, **18**, 593–603. [40](#), [55](#), [57](#)
- Li, Z. (2006). *Contribution à l’élaboration d’algorithmes d’isolation et d’identification de défauts dans les systèmes non linéaires*. Phd thesis, Institut National des Sciences Appliquées, Toulouse (France). [165](#)
- Li, Z. & Dahhou, B. (2006). An observers based fault isolation approach for non-linear dynamic systems. In *2nd International Symposium on Communications, Control and Signal Processing (ISCCSP’2006)*, Marrakech (Morocco). [165](#)

- Li, Z. & Dahhou, B. (2007). Fault isolation for nonlinear dynamic systems based on parameter intervals. *International Journal of Systems Science*, **38**, 531 – 547. [128](#), [129](#), [165](#), [166](#), [170](#)
- Li, Z., Dahhou, B. & Roux, G. (2006). A fault isolation method for nonlinear dynamic systems based on monotonous observers. In *Multiconference on Computational Engineering in Systems Applications (CESA'2006)*, Beijing (China). [165](#)
- Lin, Y. & Stadherr, M. (2007). Fault detection in continuous-time systems with uncertain parameters. In *American Control Conference, 2007*, 3216–3221. [61](#), [175](#)
- Lin, Y. & Stadherr, M. (2008). Fault detection in nonlinear continuous-time systems with uncertain parameters. *AIChE Journal*, **54**, 2335–2345. [61](#), [175](#)
- Ljung, L. & Glad, T. (1994). *Modeling of Dynamic Systems*. Prentice Hall. [113](#)
- Manders, E. (2003). *A combined statistical detection and qualitative fault isolation scheme for abrupt faults in dynamic systems*. Ph.D. thesis, Vanderbilt University, Nashville (USA). [124](#)
- Manders, E. (2008). *Diagnostic et approches ensemblistes à base des zonotopes*. Ph.D. thesis, Université de Cergy-Pontoise, Cergy-Pontoise (France). [61](#)
- Manders, E. & Biswas, G. (2003). FDI of abrupt faults with combined statistical detection and estimation and qualitative fault isolation. In *5th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes SAFEPROCESS*, 347–352, Washington. [174](#)
- Manders, E., Narasimhan, S., Biswas, G. & Mosterman, P. (2000). A combined qualitative/quantitative approach for fault isolation in continuous dynamic systems. In *4th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes SAFEPROCESS*, 1074–1079, Budapest, Hungary. [103](#), [128](#), [129](#)

- 
- Milanese, M. & Vicino, A. (1991). Optimal estimation theory for dynamic systems with set membership uncertainty: An overview. *Automatica*, **27**, 997–1009. 25
- Milanese, M., Norton, J., Piet-Lahanier, H. & Walter, É. (1996). *Bounding approaches to system identification*. Plenum Publishing Corporation, New York. 25, 128
- MONET (1998). European network of excellence on model based systems and qualitative reasoning. <http://monet.aber.ac.uk:8080/monet/index.html>. 17
- Moore, R.E. (1966). *Interval analysis*. Prentice-Hall. 60
- Mosterman, P.J. & Biswas, G. (1999). Diagnosis of continuous valued systems in transient operating regions. *IEEE Trans. on Systems, Man and Cybernetics, Part A*, **29**, 554–565. 85, 88, 103, 114, 115
- Neumaier, A. (2004). Complete search in continuous global optimization and constraint satisfaction. *Acta Numerica*, **13**, 271–369. 74
- Nguyen, D. & Blanke, M. (Submitted). Fault-tolerant positioning control for offshore vessels actuated by thruster and mooring systems. 138
- Nguyen, D., Blanke, M. & Sørensen, A. (2007). Diagnosis and fault-tolerant control for thruster-assisted position mooring system. In *6th IFAC Conference on Control Applications in Marine Systems*, Bol, Croatia. 138
- Niemann, H. (2006). A setup for active fault diagnosis. *IEEE Trans. on Automatic Control*, **51**, 1572–1578. 50
- Niemann, H., Saberi, A., Stoorvogel, A. & Sannuti, P. (1999). Exact, almost and delayed fault detection: an observer based approach. *Int. J. Robust Nonlinear Control*, **9**, 215–238. 174
- Norton, J. (1994). *Modeling Techniques for Uncertain Systems*, chap. Prior Information and Other Modelling Constraints for Bounded-Parameter Models, 43–51. Birkhäuser. 25

## REFERENCES

---

- Nyberg, M. (1999). *Model Based Fault Diagnosis: Methods, Theory, and Automotive Engine Applications*. Ph.D. thesis, Linköpings Universitet. [14](#)
- Nyberg, M. (2002). Model-based diagnosis of an automotive engine using several types of fault models. *IEEE Transactions on Control Systems Technology*, **10**, 679–689. [129](#), [146](#)
- Nyberg, M. & Nielsen, L. (1998). Model based diagnosis for the air intake system of the SI-engine. *SAE Transactions, Journal of Commercial Vehicles*, **106**, 9–20. [77](#), [146](#)
- Nyberg, M., Stutte, T. & Wilhelmi, V. (2001). Model based diagnosis of the air path of an automotive diesel engine. IFAC Workshop: Advances in Automotive Control, 629–634, Karlsruhe, Germany. [146](#)
- Ocampo-Martínez, C., Tornil, S. & Puig, V. (2006). Robust fault detection using interval constraints satisfaction and set computations. In *6th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes SAFEPROCESS*. [61](#)
- Olive, X., Travé-Massuyès, L. & Thomas, J. (2003). Complementing an interval based diagnosis method with sign reasoning in the automotive domain. In *5th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes SAFEPROCESS*, 615–620. [102](#)
- Orlov, A. (1991). How often are the observations normal? *Industrial laboratory*, **57**, 770–772. [99](#)
- Palade, V., Bocaniala, C. & Jain, L., eds. (2006). *Computational Intelligence in Fault Diagnosis*. Springer Verlag. [21](#)
- Patton, R. & Chen, J. (1991). A re-examination of the relationship between parity space and observer-based approaches in fault diagnosis. *European Journal of Diagnosis and Safety in Automation (Revue Européenne Diagnostic et Sûreté de fonctionnement)*, **1**, 183–200. [23](#)

## REFERENCES

---

- Patton, R.J. (1997). Robustness in model-based fault diagnosis: the 1997 situation. *Annual Reviews in Control*, **21**, 103–123. [24](#)
- Patton, R.J., Frank, P.M. & Clark, R.N. (2000). *Issues of fault diagnosis for dynamic systems*. Springer, London. [12](#), [20](#)
- Pearson Longman (2008). Longman Dictionary of Contemporary English. <http://www.pearsonlongman.com/>. [9](#)
- Ploix, S. & Gentil, S. (2000). Causal strategy for set-membership fault diagnosis. In *Proceedings of the 39th IEEE Conference on Decision and Control*, vol. 3, 2627–2632. [60](#)
- Ploix, S., Désinde, M. & Touaf, S. (2005). Automatic design of detection tests in complex dynamic systems. In *16th IFAC World Congress*, Prague. [31](#)
- Pouliezos, A.D. & Stavrakakis, G.S. (1994). *Real Time Fault Monitoring of Industrial Processes*. Kluwer Academic Publisher, The Netherlands. [22](#), [23](#)
- Poulsen, N. & Niemann, H. (2007). Stochastic change detection based on an active fault diagnosis approach. In *46th IEEE Conference on Decision and Control*, 346–351. [50](#)
- Puig, V., Quevedo, J., Escobet, T. & Pulido, B. (2004). On the integration of fault detection and isolation in model-based fault diagnosis. In *15th International Workshop on Principles of Diagnosis DX*, 227–232. [103](#)
- Puig, V., Schmid, F., Quevedo, J. & Pulido, B. (2005). A new fault diagnosis algorithm that improves the integration of fault detection and isolation. In *44th IEEE Conference on Decision and Control, and the European Control Conference*, Sevilla. [102](#)
- Puig, V., Stancu, A., Escobet, T., Nejjari, F., Quevedo, J. & Patton, R.J. (2006). Passive robust fault detection using interval observers: Application to the DAMADICS benchmark problem. *Control Engineering Practice*, **14**, 621–633. [76](#)

- Puig, V., Quevedo, J., Escobet, T., Nejjari, F. & de las Heras, S. (2008). Passive robust fault detection of dynamic processes using interval models. *IEEE Transactions on Control Systems Technology*, **16**, 1083–1089. [61](#)
- Pulido, B. & Alonso-González, C. (2004). Possible conflicts: A compilation technique for consistency-based diagnosis. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, **34**, 2192–2206. [17](#), [31](#), [55](#)
- Pulido, B. & Gelso, E. (2005). Viabilidad de una técnica de compilación de dependencias para diagnosis basada en modelos en tareas de supervisión. In *XI Conferencia de la Asociación Española para la Inteligencia Artificial, CAEPIA'05*, Santiago de Compostela, Spain. [55](#)
- Rajaraman, S. (2005). *Robust Model-Based Fault Diagnosis Of Chemical Process Systems*. Phd thesis, Texas A&M University, Texas (USA). [165](#)
- Reiter, R. (1987a). *Readings in Model-Based Diagnosis*, chap. A theory of diagnosis from first principles. Morgan Kaufmann. [17](#)
- Reiter, R. (1987b). A theory of diagnosis from first principles. *Artificial Intelligence*, **32**, 57–95. [17](#), [18](#)
- Rinner, B. & Weiss, U. (2004). Online monitoring by dynamically refining imprecise models. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, **34**, 1811–1822. [60](#), [66](#)
- Roychoudhury, I., Biswas, G. & Koutsoukos, X. (2006). A bayesian approach to efficient diagnosis of incipient faults. In *17th International Workshop on Principles of Diagnosis DX*, 243–264. [85](#), [87](#), [115](#), [131](#)
- Roychoudhury, I., Biswas, G. & Koutsoukos, X. (2009). Designing distributed diagnosers for complex continuous systems. *IEEE Transactions on Automation Science and Engineering*, To appear. [117](#)
- Samantaray, A. & Ould Bouamama, B. (2008). *Model-based Process Supervision: A Bond Graph Approach*. Springer Verlag. [14](#), [16](#)

- 
- Samantaray, A., Ghoshal, S., Chakraborty, S. & Mukherjee, A. (2005). Improvements to single-fault isolation using estimated parameters. *Simulation*, **81**, 827–845. [128](#)
- Shary, S. (2002). A new technique in systems analysis under interval uncertainty and ambiguity. *Reliable Computing*, **8**, 321–418. [71](#)
- SIGLA/X group, . (1999). Modal intervals. Basic tutorial. In *Proceedings of MISC 99. Applications of Interval Analysis to Systems and Control*, 157–227, Universitat de Girona, Spain. [60](#), [63](#)
- Simani, S., Fantuzzi, Z. & Patton, R. (2003). *Model-based fault diagnosis in dynamic systems using identification techniques*. Springer Verlag. [14](#), [20](#), [24](#), [128](#)
- Stancu, A., Puig, V. & Quevedo, J. (2003). Gas turbine model-based robust fault detection using a forward - backward test. In *Second International Workshop on Global Constraint Optimization and Constraint Satisfaction, COCOS 2003*, 154–170. [27](#), [61](#)
- Staroswiecki, M. (2002). *Encyclopedia of Life Support Systems*, chap. Structural analysis for fault detection and isolation and for fault tolerant control. Fault Diagnosis and Fault Tolerant Control, Oxford, UK. [30](#), [32](#), [35](#)
- Staroswiecki, M. & Comtet-Varga, G. (2001). Analytical redundancy relations for fault detection and isolation in algebraic dynamic systems. *Automatica*, **37**, 687–699. [32](#)
- Struss, P. (2008). *Handbook of Knowledge Representation*, chap. 10: Model-based Problem Solving, 395–465. Elsevier. [17](#)
- Svärd, C. & Nyberg, M. (2008). A mixed causality approach to residual generation utilizing equation system solvers and differential-algebraic equation theory. In *19th International Workshop on Principles of Diagnosis DX*, 181–188, Blue Mountains, Australia. [46](#)
- Travé-Massuyès, L. & Pons, R. (1997). Causal ordering for multiple mode systems. In *Eleventh International Workshop on Qualitative Reasoning (QR-97)*, 203–214, Cortona, Italia. [54](#)

- Travé-Massuyès, L., Escobet, T., Pons, R. & Tornil, S. (2001). The Ca En diagnosis system and its automatic modelling method. *Computación y Sistemas*, **5**, 128–143. [27](#)
- Travé-Massuyès, L., Escobet, T. & Olive, X. (2006). Diagnosability analysis based on component supported analytical redundancy relations. *IEEE Trans. on Systems, Man and Cybernetics, Part A : Systems and Humans*, **36**. [31](#)
- Venkatasubramanian, V., Rengaswamy, R. & Kavuri, S. (2003a). A review of process fault detection and diagnosis: Part II: Qualitative models and search strategies. *Computers and Chemical Engineering*, **27**, 313–326. [14](#)
- Venkatasubramanian, V., Rengaswamy, R., Yin, K. & Kavuri, S. (2003b). A review of process fault detection and diagnosis: Part I: Quantitative model-based methods. *Computers and Chemical Engineering*, **27**, 293–311. [14](#)
- Venkatasubramanian, V., Rengaswamy, R., Yin, K., Kavuri, S. & Yin, K. (2003c). A review of process fault detection and diagnosis: Part III: Process history based methods. *Computers and Chemical Engineering*, **27**, 327–346. [14](#)
- Verde, C., Gentil, S. & Sanchez Parra, M. (2007). Invariability of the residual structure with respect to perfect matching. In *Proceedings of 3rd IFAC Symposium on System, Structure and Control SSSC07*, IFAC, Foz do Iguaçu (Brazil). [32](#)
- Walter, É. & Pronzato, L. (1997). *Identification of parametric models from experimental data*. Communications and control engineering series, Springer, Paris. [128](#)
- Watkins, J. & Yurkovich, S. (1996). Fault detection using set-membership identification. In *Proceedings of IFAC World Congress*, 61–66, Sydney. [27](#), [61](#)
- Wiegand, M. (1991). *Constructive qualitative simulation of continuous dynamic systems*. Ph.D. thesis, Heriot-Watt University, Edinburgh, Scotland. [26](#)
- Witczak, M. (2003). *Identification and Fault Detection of Non-Linear Dynamic Systems*, vol. 1 of *Lecture Notes in Control and Computer Science*. University of Zielona Góra Press, Poland. [11](#)



## REFERENCES

---

Zhang, X. (1989). *Auxiliary signal design in fault detection and diagnosis*. Springer Verlag. 50